NBS-GCR-83-451

Computer Fire Code VI - Volume 2

December 1983

Sponsored by
U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Fire Research
Washington, DC 20234

NBS-GCR-83-451

COMPUTER FIRE CODE VI - VOLUME 2

J. B. Gahm

Harvard University
Division of Applied Sciences
Cambridge, MA

December 1983

Sponsored by
U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Fire Research
Washington, DC 20234

NOTICE

This report was prepared for the Center for Fire Research of the National Engineering Laboratory, National Bureau of Standards under Grant Number NB82NADA3030. The statements and conclusions contained in this report are those of the authors and do not necessarily reflect the views of the National Bureau of Standards or the Center for Fire Research.

TABLE OF CONTENTS

Appendix D

Part 1 - BINPS

Part 2 - DBLES

Part 3 - Common Blocks

PART 1 - BINPS

```
      PROGRAM BINP
      IMPLICIT REAL*8 (A-H,O-Z)
C
C THIS PROGRAM CREATES AN INPUT FILE FOR THE MAIN FIRE CODE 6
C SO THAT THE PROGRAM CAN BE RUN IN BATCH MODE.
C
C
C INITIALIZES VARIABLES FOR USE IN THE COMPUTER FIRE CODE PACKAGE,
C
C CALLS INPUT3 TO INPUT DATA FROM THE TERMINAL.
C
C
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,IC6.CMN
$INCLUDE 9,BLDNG.CMN
      INTEGER O,R,W,V
      PARAMETER (PI=3.1415927D0)
      PARAMETER (IBATCH=0)
C
      CHARACTER PROSE*80,ICHAR*1
      EQUIVALENCE (PROSE(1:1),ICHAR)
      INTEGER INPSIZ
      PARAMETER (INPSIZ=5380/4)
      INTEGER IDAT(INPSIZ)
      EQUIVALENCE (ZBVZZ,IDAT)
C
C OPEN OUTPUT AND INPUT FILES

C FOR USE ON HARVARD VAX 11/780 ONLY
C


 8880 WRITE(6,8888)

 8888 FORMAT(' IS THERE AN ALREADY EXISTING DATA FILE WHICH YOU WISH',/,
     1  ' TO MODIFY? (Y/N)    ',/)

 8890 READ(5,8892,ERR=8880) ICHAR
 8892 FORMAT(A1)
      IF (ICHAR .EQ. 'N') GOTO 8950

      IF (ICHAR .NE. 'Y') GOTO 8880
 8894 WRITE(6,8896)

 8896 FORMAT (' PLEASE ENTER THE NAME OF THE DATA FILE WHICH YOU WISH ',/,
     1  ' TO MODIFY.    DEFAULT IS "SR.DAT"   ',//)
 8898 READ(5,8899,ERR=8894) PROSE(1:24)


 8899 FORMAT(6A4)
      IF(PROSE(1:2) .EQ. '  ') PROSE(1:24) = 'SR.DAT                  '
      OPEN (UNIT=33,FILE=PROSE(1:24),STATUS='OLD')
      GOTO 10
 8950 WRITE(6,8952)

 8952 FORMAT (' PLEASE ENTER THE NAME OF THE DATA FILE YOU ARE ',/,
     1  ' ABOUT TO CREATE           ',/)
 8954 READ(5,8956,ERR=8950) PROSE(1:24)
 8956 FORMAT(6A4)
      OPEN (UNIT=19,FILE = PROSE(1:24),STATUS='NEW')

C GET THE DATA FROM THE INPUT FILE TO BE EDITED.
```

```
   10 DO 15 I = 1,INPSIZ
      READ(33) IDAT(I)
   15 CONTINUE
C
C OUTPUT MESSAGE ABOUT INPUT ERRORS TO THE EFFECT THAT
C INPUT ERRORS CAUSE ADDITIONAL READS TO BE EXECUTED.
C
      WRITE (IWTTY,290)
C
C SELECT PHYSICS SUBROUTINES
   30 CALL SELSUB
C
C INPUT ROOM SETUP AND GEOMETRIES
   40 WRITE (IWTTY,200)
      READ (IRTTY,190) ICHAR
      IF (ICHAR .EQ. 'N' .OR. ICHAR .EQ. ' ') GOTO 60
      IF (ICHAR .NE. 'Y') GOTO 40
   50 CALL INPUT3
C
C RECAPITULATE INPUT PARAMETERS ON THE TERMINAL SCREEN FOR
C VERIFICATION, IF INTERACTIVE MODE .AND. GEOMETRIC CHANGES WERE MADE
C (I.E., IF ANSWER ABOVE WAS 'Y' AND INPUT3 WAS CALLED)
   60 IF (IBATCH .EQ. 0) THEN
        WRITE (IWTTY,250)
        CALL RECAP (IWTTY)
        IF (IASKO2(0) .NE. 0) GOTO 50
      END IF
CC
   70 IFNOUT = 0
C       IF (IBATCH .EQ. 0) THEN
C    808 WRITE (IWTTY,902)
C         READ (IRTTY,900) ICHAR
C         IF (ICHAR .EQ. 'N' .OR. ICHAR .EQ. ' ') GOTO 811
C         IF (ICHAR .NE. 'Y') GOTO 808
C    809 IFNOUT = 1
C
C ASK USER FOR OUTPUT INTERVAL CONTROL PARAMETERS, OR
C DEFAULT TO THE LISTED VALUES
C    80   WRITE (IWTTY,350)'TTY OUTPUT INTERVAL','20.0'
C          READ (IRTTY,240,ERR=80) TTYOUT
C          IF (TTYOUT .LE. 0.D0) TTYOUT = 20.D0
C        ELSE
C          IFNOUT = 1
C          WRTTY = 1.D+4
C        END IF
C    90 WRITE (IWTTY,350) 'DISK OUTPUT INTERVAL','10.0'
C        READ (IRTTY,240,ERR=90) DSKOUT
C        IF (DSKOUT .LE. 0.D0) DSKOUT = 10.D0
C
C ASK USER FOR LOOP CONTROL PARAMS, OR DEFAULT
C   100 WRITE (IWTTY,350) 'RUN LENGTH          ','500.0'
C        READ (IRTTY,240,ERR=100) FINISH
C        IF (FINISH .LE. 0.D0) FINISH = 500.D0
C        WRITE (IWDSK) TTYOUT,DSKOUT,FINISH
      DO 9567 I=1,INPSIZ
      WRITE(IWDSK) IDAT(I)
 9567 CONTINUE
      CLOSE(IWDSK)
      STOP
C
  190 FORMAT(A1)
  200 FORMAT(/,'$CHANGE FROM STANDARD GEOMETRIC AND PHYSICAL',
    1         ' PARAMETERS? (Y/N): ')
  210 FORMAT('0DO YOU WANT OUTPUT IN COLUMNAR SHORT FORM ',
     1       '(REQUIRES LINE PRINTER) INSTEAD OF',/,
     2       '$(DEFAULT) TABULAR LONG FORM? (Y/N): ')
  220 FORMAT('0DO YOU WANT THE TERMINAL OUTPUT IN ',
     1      'GRAPHICAL FORM INSTEAD OF (OR IN',/,
     2      '$ADDITION TO THE DEFAULT) TABULAR FORM? ',
     3      '(Y/N): ')
C  902 FORMAT(/,'$SUPPRESS HNORM AND FNORM OUTPUT? (Y/N): ' )
  230 FORMAT (/,'$IS THIS BATCH MODE? (Y/N): ')
  240 FORMAT(F12.4)
C
  250 FORMAT (/,T6,'RECAPITULATION OF INPUT FOR THIS RUN:',/)
  260 FORMAT (/,T10,'GEOMETRIC AND PHYSICAL PARAMETERS:',/)
  270 FORMAT (' ---------------',/,' FILES IN USE:')
  280 FORMAT('0TYPE IN COMMENT LINES FOR LABEL, ENDING WITH',
     1       ' AN EXTRA <RET>')
  290 FORMAT('1IN THE INPUT SEQUENCE TO FOLLOW, NOTE THAT',
     1       ' AN INPUT ERROR'/' -- E.G. TYPING A LETTER WHERE A',
     2       ' DIGIT IS REQUIRED --'/' CAUSES THE QUESTION TO BE',
     3       ' REASKED.'/,
     4       ' THE SYMBOL <RET> STANDS FOR CARRIAGE RETURN, OR',
     5       ' NEWLINE, OR'/' WHATEVER KEY ON THE TERMINAL IS USED',
     6       ' TO ENTER DATA.'//)
  300 FORMAT ('$PROGRAMMER-RUN NUMBER = ')
  310 FORMAT(' RUN ',A80,/,' ---------------')
  320 FORMAT(A80)
  330 FORMAT('    ',A80)
  340 FORMAT(A30)
  350 FORMAT('0INPUT DESIRED ',A20,' (IN F FORMAT), OR <RET> TO',/,
     1       '$DEFAULT TO ',A6,' SECONDS: ')
      END
      SUBROUTINE COPINP(ICOP)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CODED BY M.SPIVAKOVSKY, JUNE 1980.
C
C COPIES THE INPUT DATA FROM ANOTHER
C ROOM (IF ICOP = 1)
C OBJECT - GEOMETRIC PROPERTIES (IF ICOP = 2)
C OBJECT - PHYSICAL PROPERTIES (IF ICOP = 5)
C VENT (IF ICOP = 3)
C OR WALL (IF ICOP = 4).
C
$INCLUDE 9/DIMN.CMN
$INCLUDE 9/IO6.CMN
$INCLUDE 9/BLDNG.CMN
C
      COMMON /INPUTS/ KR, KO, KV, KW,
     +                ITEM(25), VAL1(25), VAL2(25), INCNT
C
      IF (ITEM(INCNT) .NE. 100) RETURN
      GO TO (10,20,30,40,25),ICOP
C
   10 WRITE (IWTTY,200) 'ROOM '
      READ (IRTTY,100,ERR=10) KC
      ZLRZY(KR) = ZLRZY(KC)
      ZLRZX(KR) = ZLRZX(KC)
      ZHRZZ(KR) = ZHRZZ(KC)
      OBJS(KR) = OBJS(KC)
      ZKAZZ(KR) = ZKAZZ(KC)
      RETURN
C
   20 WRITE (IWTTY,201)
  201 FORMAT(X,'ENTER THE OBJECT AND ROOM NUMBER OF THE OBJECT
     1 YOU WANT TO COPY FROM.')
      READ (IRTTY,101,ERR=20) KC,KC1
  101 FORMAT(2I3)
      ZXCZZ(KC,KR)=ZXCZZ(KC,KC1)
      ZYCZZ(KC,KR)=ZYCZZ(KC,KC1)
      ZHCZZ(KC,KR)=ZHCZZ(KC,KC1)
      ZNCZZ(KC,KR)=ZNCZZ(KC,KC1)
      ZRFZO(KC,KR)=ZRFZO(KC,KC1)
      ZRFZI(KC,KR)=ZRFZI(KC,KC1)
      ZRFZM(KC,KR)=ZRFZM(KC,KC1)
      RETURN
C
   25 WRITE (IWTTY,202)
  202 FORMAT(X,'ENTER THE OBJECT AND ROOM NUMBER OF THE OBJECT YOU
     1 WANT TO COPY FROM.')
      READ (IRTTY,101,ERR=25) KC,KC1
      STATE(KC,KR)=STATE(KC,KC1)
      FIRTYP(KC,KR) = FIRTYP(KC,KC1)
      XGAMMA(KC,KR)=XGAMMA(KC,KC1)
      XGAMAS(KC,KR)=XGAMAS(KC,KC1)
      ZJCZZ(KC,KR)=ZJCZZ(KC,KC1)
      ZCCZZ(KC,KR)=ZCCZZ(KC,KC1)
      VMCZZ(KC,KR)=VMCZZ(KC,KC1)
      EB(KC,KR)=EB(KC,KC1)
      CHI(KC,KR)=CHI(KC,KC1)
      QF(KC,KR)=QF(KC,KC1)
      QVAP(KC,KR)=QVAP(KC,KC1)
      ZMCZO(KC,KR)=ZMCZO(KC,KC1)
      ZKCIG(KC,KR)=ZKCIG(KC,KC1)
      ZKCPY(KC,KR)=ZKCPY(KC,KC1)
      FCO2(KC,KR)=FCO2(KC,KC1)
      FCO(KC,KR)=FCO(KC,KC1)
      FS(KC,KR)=FS(KC,KC1)
      FH2O(KC,KR)=FH2O(KC,KC1)
      AFIRE(KC,KR)=AFIRE(KC,KC1)
      TMFGZ(KC,KR)=TMFGZ(KC,KC1)
      RETURN
C
   30 WRITE (IWTTY,200) 'VENT '
      READ (IRTTY,100,ERR=30) KC
      ZBVZZ(KV)=ZBVZZ(KC)
      ZHVZZ(KV)=ZHVZZ(KC)
      ZHTZZ(KV)=ZHTZZ(KC)
      VLINK(1,KV) = VLINK(1,KC)
      VLINK(2,KV) = VLINK(2,KC)
      RETURN
C
   40 WRITE (IWTTY,200) 'WALL '
      READ (IRTTY,100,ERR=40) KC
      ZNWZZ(KW)=ZNWZZ(KC)
      ZJWZZ(KW)=ZJWZZ(KC)
      ZCWZZ(KW)=ZCWZZ(KC)
      VMWZZ(KW)=VMWZZ(KC)
      RETURN
C
  100 FORMAT (I1)
  200 FORMAT ('$ENTER INDEX NUMBER OF ',A6,' TO BE COPIED: ')
      END
      SUBROUTINE DISP(IWHAT,*)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALLED BY INPUT3.
```

```
C
C IN FILE IO5281.FOR
C
C
C PRINTS TABLE OF INPUT VALUES ON THE TERMINAL
C
C      IWHAT=1: VARIABLES INDEXED BY ROOM (KR);
C      IWHAT=2: VARIABLES INDEXED BY OBJECT (KO,KR), GEOMETRY;
C      IWHAT=3: VARIABLES INDEXED BY VENT (KV);
C      IWHAT=4: VARIABLES INDEXED BY WALL (KW);
C      IWHAT=5: VARIABLES NOT INDEXED BY ANYTHING;
C      IWHAT=6: VARIABLES INDEXED BY OBJECT (KO,KR), PHYSICS.
C
C      * ARGUMENT: SECOND RETURN ADDRESS ("RETURN 1") -- DISP RETURNS TO
C      THIS LABEL IN INPUT3 IF NO CHANGES TO BE MADE TO THE TABLE.
C
C      IFLAG IS AN INTERNAL FLAG. IF =0, THE TABLE RELEVANT TO IWHAT IS
C      DISPLAYED FOR THE FIRST TIME; IF NO CHANGES ARE TO BE MADE
C      (USER PICKS ITEM # 0), WE USE ALTERNATE RETURN AND GO TO NEXT
C      TABLE CALL IN INPUT3.  OTHERWISE, IFLAG IS SET TO 1 AND RETURN
C      TO INPUT3 DOES REGULAR READING TILL NULL ENTRY IS FOUND, AND
C      THEN THE SAME CALL TO DISP (SAME IWHAT TABLE)
C      IS MADE -- DISP SEES IFLAG IS NONZERO AND THEN DOES A VERIFY
C      ROUTINE.  IF THEN ALL VALUES ARE OK, IFLAG IS ZEROED AND WE
C      ALTERNATE-RETURN AS ABOVE; ELSE, RETURN TO SAME READING OF
C      ITEM(INCNT) IN INPUT3 AS ABOVE.
C
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,BLDNG.CMN
$INCLUDE 9,IO6.CMN
C
      COMMON /INPUTS/ KR, KO, KV, KW,
     1                ITEM(25), VAL1(25), VAL2(25), INCNT
C
      DATA IFLAG, IFIRST/ 0, 0 /
C
      IF (IBATCH .NE. 0) THEN
        IF (IFLAG .EQ. 0) GOTO 100
        GOTO 130
      END IF
C
      WRITE (IWTTY,160)
      IF (IFIRST .EQ. 0) WRITE (IWTTY,170)
      IFIRST = IFIRST+1
   10 GO TO (20,30,50,60,70,40) IWHAT
C
   20 WRITE (IWTTY,180) KR, ZLRZX(KR), ZLRZY(KR),
     1       ZHRZZ(KR), OBJS(KR), ZKAZZ(KR)
      GOTO 80
   30 WRITE(IWTTY,190) KR,KO,KO,ZXOZZ(KO,KR),ZYOZZ(KO,KR),
     1       ZHOZZ(KO,KR),ANGH(KO,KR),ANGV(KO,KR),ZNOZZ(KO,KR),
     2       ZRFZO(KO,KR),ZRFZI(KO,KR),ZRFZM(KO,KR)
      WRITE (IWTTY,200)
      GOTO 80
   40 CALL DSPASK(IFLAG)
      GOTO 80
   50 WRITE (IWTTY,210) KV, ZBVZZ(KV), ZHVZZ(KV), ZHTZZ(KV),
     1       (VLINK(I,KV),I=1,2)
      GOTO 80
   60 WRITE (IWTTY,220) KW, WLINK(1,KW), WLINK(2,KW), ZNWZZ(KW),
     1        ZJWZZ(KW), ZCWZZ(KW), VMWZZ(KW)
      GOTO 80
   70 WRITE (IWTTY,230) CP, 1.00/ZLAMDA, ZKACUT, ZOWZM, ZOWZN,
      1       ZCAZZ, ALPHA, CD
C
   80 IF (IFLAG .NE. 0) GOTO 120
   90 WRITE (IWTTY,140)
  100 INCNT=0
  110 INCNT=INCNT+1
      READ (IRTTY,150,ERR=10) ITEM(INCNT),VAL1(INCNT),VAL2(INCNT)
      IF (ITEM(INCNT) .NE. 0) GOTO 110
      IF (INCNT .EQ. 1) RETURN 1
      INCNT=0
      IFLAG=IFLAG+1
      RETURN
  120 IF (IASK02(0) .NE. 0) GOTO 90
  130 IFLAG=0
      RETURN 1
C
  140 FORMAT (' TO CHANGE ITEMS, ENTER "ITEM,VALUE<RET>" ',
     1         '(I3,F12.4); END WITH AN EXTRA <RET>:')
  150 FORMAT (I3,2(E15.4))
  160 FORMAT(24(/))
  170 FORMAT (' IN EACH TABLE BELOW, IF ALL THE VALUES ARE',
     1         ' SATISFACTORY, THEN AS ',/,' INDICATED, ENTER ITEM 0',
     2         ' AND <RET>.',
     3         '  IN ORDER TO CHANGE ANY (OR ALL)'/' OF THE VALUES,',
     4         ' TYPE IN THE APPROPRIATE ITEM NUMBER IN I3 FORMAT,',
     5        /' THEN A COMMA, AND THEN THE RELEVANT DATA VALUE',
     6         '  IN F12.4 OR E12.4,'/' FOLLOWED BY <RET>.',
     7         ' REPEAT FOR AS MANY ITEMS AS DESIRED, ENDING'/' WITH',
     8       ' AN EXTRA <RET>. (E.G.: "4,3.<RET>5,293.<RET><RET>',//)
  180 FORMAT(' ROOM NUMBER ',I2,':',//,
     1           T4,'0) CHANGE NOTHING (CONTINUE)',
     2          /,T4,'1) LENGTH ALONG X:',T35,1PE11.4,
     3          /,T4,'2) LENGTH ALONG Y:',T35,1PE11.4,
     4          /,T4,'3) HEIGHT:',T35,1PE11.4,
     5          /,T4,'4) # OF OBJECTS:',T35,I3,'.',
     6          /,T4,'5) AMBIENT TEMPERATURE:',T35,1PE11.4,
     7         /,T2,'100) COPY ALL ABOVE DATA FROM ANOTHER ROOM',
     8   /,'(NOTE THAT TYPING "100" ALONE SUFFICES; THERE ARE NO',
     9   /,'RELEVANT DATA VALUES NEEDED)',/)

  190 FORMAT (' ROOM NUMBER ',I2,', OBJECT NUMBER ',I2,
     1          ' (OBJ NO = ',I2,') - GEOMETRIC PROPERTIES:',
     2          /,T4,'0) CHANGE NOTHING (CONTINUE)',
     3          /,T4,'1) X-COORDINATE:',T55,1PE11.4,
     4          /,T4,'2) Y-COORDINATE:',T55,1PE11.4,
     5          /,T4,'3) HEIGHT:',T55,1PE11.4,
     6          /,T4,'4) ANGLE WITH HORIZONTAL (IN DEGREES):',T55,1PE11.4,
     7          /,T4,'5) IF SURFACE IS VERTICAL,',
     8                ' ANGLE WITH XZ-PLANE:',T55,1PE11.4,
     9          /,T4,'6) THICKNESS:',T55,1PE11.4,
     1          /,T4,'7) INITIAL BURNING RADIUS:',T55,1PE11.4,
     2          /,T4,'8) OBJECT RADIUS:',T55,1PE11.4,
     3          /,T4,'9) MAXIMUM BURNING RADIUS:',T55,1PE11.4,
     4    /,T3,'10) INSTEAD OF ITEMS 8 & 9, INPUT DIMENSIONS OF ',
     5         'RECTANGULAR OBJECT',
     6         /,T7,'(ITEM NO., LENGTH AND WIDTH VALUES ALL ON ONE LINE: '
     7         'E.G., "10,1.5,3.85")',
     8          /,T2,'100) COPY ALL ABOVE DATA FROM ANOTHER OBJECT')
  200 FORMAT (/,T5,'UNLESS THERE IS A COMPELLING REASON TO CHOOSE',
     1          ' OTHERWISE, THE LOCATION',
     2          /,T5,'OF AN OBJECT IS THE POINT AT THE GEOMETRICAL',
     3          ' CENTER OF THE PRINCIPAL',
     4          /,T5,'EXPOSED SURFACE OF THE OBJECT.',/)
  210 FORMAT(' VENT NUMBER ',I2,/
     1          /,T4,'0) CHANGE NOTHING (CONTINUE)',
     2          /,T4,'1) WIDTH:',T30,1PE11.4,
     3          /,T4,'2) HEIGHT:',T30,1PE11.4,
     4          /,T4,'3) TRANSOM DEPTH:',T30,1PE11.4,
     5          /,T4,'4) SIDE 1 IS IN ROOM:',T30,I2,
     6          /,T4,'5) SIDE 2 IS IN ROOM:',T30,I2,
     7          /,T2,'100) COPY ALL ABOVE DATA FROM ANOTHER VENT')
  220 FORMAT (' WALL NUMBER ',I2,' (SIDE 1 IN ROOM ',I2,
     1          '; SIDE 2 IN ',I2,')',/,
     2          /,T4,'0) CHANGE NOTHING (CONTINUE)',
     3          /,T4,'1) THICKNESS:',T30,1PE11.4,
     4          /,T4,'2) THERMAL CONDUCTIVITY:',T30,1PE11.4,
     5          /,T4,'3) SPECIFIC HEAT:',T30,1PE11.4,
     6          /,T4,'4) DENSITY:',T30,1PE11.4,
     7          /,T2,'100) COPY ALL ABOVE DATA FROM ANOTHER WALL')
  230 FORMAT (' NON-INDEXED VARIABLES:',/,
     1          /,T4,'0) CHANGE NOTHING (CONTINUE)',
     2          /,T4,'1) SPECIFIC HEAT OF AIR:',T50,1PE11.4,
     3          /,T4,'2) ABSORPTION COEFFICIENT OF FLAME:',T50,1PE11.4,
     4          /,T4,'3) AMBIENT TEMP. OUTSIDE BUILDING:',T50,1PE11.4,
     5       /,T4,'4) MAX. HEAT TRANSFER COEFFICIENT (WALLS):',T50,1PE11.4
     6       /,T4,'5) MIN. HEAT TRANSFER COEFFICIENT (WALLS):',T50,1PE11.4
     7          /,T4,'6) HEAT TRANSFER COEFFICIENT (OBJECTS):',T50,1PE11.4
     8          /,T4,'7) PLUME ENTRAINMENT COEFFICIENT:',T50,1PE11.4,
     9          /,T4,'8) VENT FLOW COEFFICIENT:',T50,1PE11.4,/)
      END
      SUBROUTINE DSPASK(IFLAG)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C IN FILE IO5281.FOR
C CALLED BY DISP
C M.RABUZZI, MAY 81.
C
C CALLED TO ASK QUESTIONS FOR DISP(6,*): PHYSICS.
C
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,BLDNG.CMN
$INCLUDE 9,IO6.CMN
C
      COMMON /INPUTS/ KR, KO, KV, KW,
     1                ITEM(25), VAL1(25), VAL2(25), INCNT
C
      WRITE (IWTTY,30) KR,KO,KO
      IF (IFLAG .NE. 0) GOTO 20
C
C ASK QUESTIONS OF USER (IASKO2, IN FILE W,
C  ASKS IF OBJECT BURNING (QUERY 9); IF NOT, QUERY 10 ASKS IF FLAMMABLE)
      IFLG1 = 0
      IF (IASKO2(9) .EQ. 0) THEN
        KTYP = 0
        STATE(KO,KR) = HEATNG
        IF (IASKO2(10) .EQ. 0) IFLG1 = IFLG1+1
      ELSE
   10   WRITE (IWTTY,40)
        READ (IRTTY,50,ERR=10) KTYP
        IF (KTYP .LE. 0  .OR.  KTYP .GT. 3) GOTO 10
        FIRTYP(KO,KR) = KTYP
        STATE(KO,KR) = FLAME
        IF (KTYP .LT. 3) IFLG1 = IFLG1-1
      END IF
C DO PROPER DISPLAY OF TABLE
   20 IF (IFLG1 .NE. 0)
```

```
     1 WRITE (IWTTY,60) ZJOZZ(KO,KR),ZCOZZ(KO,KR),VMOZZ(KO,KR),
     2                       EB(KO,KR)
      IF (IFLG1 .LE. 0)
     1 WRITE (IWTTY,70) XGAMMA(KO,KR),XGAMAS(KO,KR),
     2       CHI(KO,KR),QF(KO,KR),QVAP(KO,KR),
     3       ZMOZO(KO,KR),ZKOIG(KO,KR),ZKOPY(KO,KR),FCO2(KO,KR),
     4       FCO(KO,KR), FS(KO,KR), FH2O(KO,KR)
      IF (KTYP .GT. 0) THEN
        WRITE (IWTTY,80) PSIO(KO,KR)
        IF (KTYP .EQ. 1) THEN
          WRITE (IWTTY,90) AFIRE(KO,KR)
        ELSE IF (KTYP .EQ. 3) THEN
          WRITE (IWTTY,100) TMFGZ(KO,KR)
        END IF
      END IF
      WRITE (IWTTY,110)
      RETURN
C
   30 FORMAT (' ROOM NUMBER ',I2,', OBJECT NUMBER ',I2,
     1        ' (OBJ KO = ',I2,') - PHYSICAL PROPERTIES:')
   40 FORMAT ('$INPUT TYPE OF FIRE (1=GROWING, '
     1        '2=POOL, 3=BURNER): ')
   50 FORMAT (I1)
   60 FORMAT (T4,'1) THERMAL CONDUCTIVITY:',T50,1PE11.4,
     1          /,T4,'2) SPECIFIC HEAT:',T50,1PE11.4,
     2          /,T4,'3) DENSITY:',T50,1PE11.4,
     3      /,T4,'4) EMISSIVITY/ABSORPTIVITY OF SURFACE:',T50,1PE11.4)
   70 FORMAT (T4,'5) AIR/FUEL MASS RATIO IN FREE BURN:',T50,1PE11.4,
     1          /,T4,'6) STOICHIOMETRIC MASS RATIO:',T50,1PE11.4,
     2 /,T4,'7) FRACTION OF HEAT OF COMBUSTION RELEASED:',T50,1PE11.4,
     3          /,T4,'8) HEAT OF COMBUSTION:',T50,1PE11.4,
     4          /,T4,'9) HEAT OF VAPORIZATION:',T50,1PE11.4,
     5         /,T3,'10) INITIAL FUEL MASS:',T50,1PE11.4,
     6          /,T3,'11) TEMPERATURE OF IGNITION:',T50,1PE11.4,
     7    /,T3,'12) TEMPERATURE OF ONSET OF PYROLIZATION:',T50,1PE11.4,
     8          /,T3,'13) CO2 MASS EVOLVED/FUEL MASS:',T50,1PE11.4,
     9          /,T3,'14) CO MASS EVOLVED/FUEL MASS:',T50,1PE11.4,
     1          /,T3,'15) SMOKE MASS EVOLVED/FUEL MASS:',T50,1PE11.4,
     2          /,T3,'16) WATER MASS EVOLVED/FUEL MASS:',T50,1PE11.4)
   80 FORMAT(T3,'17) SEMIAPEX ANGLE OF FLAME (IN DEGREES):',T50,1PE11.4)
   90 FORMAT(T3,'18) FIRE SPREAD PARAMETER:',T50,1PE11.4)
  100 FORMAT(T3,'19) THE GAS FLOW RATE (IN KG/SEC):',T50,1PE11.4)
  110 FORMAT(T2,'100) COPY ALL ABOVE DATA FROM ANOTHER OBJECT')
      END
      FUNCTION IASKO2(ICODE)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALLED BY DEBUG, INIT, INPUT3/DISP.
C IN FILE W05281.FOR
C
C IF REPLY TO QUERY IS 'N' OR '<RET>', FUNCTION RETURN OF ZERO;
C IF REPLY IS 'Y', FUNCTION RETURN OF +1 .
C
$INCLUDE 9,IO6.CMN
C
      CHARACTER*1 ICHAR
C
      IASKO2 = 0
   10 GOTO (20,30,40,50,60,70,80,90,100,110,120) ICODE
C     (THESE 2 LINES ARE ICODE=0, FROM IO5281 SUBRS)
      WRITE (IWTTY,150)
      GOTO 130
   20 WRITE (IWTTY,160)
```

```
      GOTO 130
   30 WRITE (IWTTY,170)
      GOTO 130
   40 WRITE (IWTTY,180)
      GOTO 130
   50 WRITE (IWTTY,190)
      GOTO 130
   60 WRITE (IWTTY,200)
      GOTO 130
   70 WRITE (IWTTY,210)
      GOTO 130
   80 WRITE (IWTTY,220)
      GOTO 130
   90 WRITE (IWTTY,230)
      GOTO 130
  100 WRITE (IWTTY,240) 'BURNING'
      GOTO 130
  110 WRITE (IWTTY,240) 'FLAMMABLE'
      GO TO 130
  120 WRITE (IWTTY,250)
      GO TO 130
  130 READ (IRTTY,140) ICHAR
      IF (ICHAR .EQ. 'N' .OR. ICHAR .EQ. ' ') RETURN
      IF (ICHAR .NE. 'Y') GOTO 10
      IASK02 = IASK02+1
      RETURN
  140 FORMAT(A1)
  150 FORMAT('$DO YOU WANT TO CHANGE YOUR INPUT? (Y/N): ')
  160 FORMAT('0DO YOU WANT "LEAVING SUBR" MESSAGES PRINTED ',
     1            'UPON EXIT'/'$FROM EACH PHYSICS SUBROUTINE? (Y/N): ')
  170 FORMAT (/'$DO YOU WANT OUTPUT EACH ITERATION? (Y/N): ')
  180 FORMAT('0WOULD YOU LIKE A LIST OF VARIABLES IN THE SYSTEM',
     1         /,' (IE: INPUT, NONZERO, & IN THE COMPUTATION MATRIX) ',
     2         /,'$PRINTED WHENEVER THE VARIABLES ARE RESCALED? (Y/N): ')
  190 FORMAT('0DO YOU WANT A LIST OF VARIABLES ENTERING',
     1         /,'$OR LEAVING THE SYSTEM AT EVERY RESCALING? (Y/N): ')
  200 FORMAT('0YOU ARE IN TROUBLE.'/,
     1       '$WOULD YOU LIKE TO TERMINATE THE RUN? (Y/N): ')
  210 FORMAT('0WOULD YOU LIKE TO SEE A SUMMARY TABLE OF PHYSICAL',/,
     1            ' VARIABLE VALUES AS COMPUTED IN THIS ITERATION?',
     2            '$(Y/N): ')
  220 FORMAT('0WOULD YOU LIKE TO SEE A SUMMARY TABLE OF PHYSICAL',/,
     1            '$VARIABLES AT THE BEGINNING OF THIS TIME STEP?',
     2            '(Y/N): ')
  230 FORMAT('$WOULD YOU LIKE TO CHANGE ANY OF YOUR DEBUGGER OPTIONS?',
     1          ' (Y/N): ')
  240 FORMAT('$IS THE OBJECT ',A9,'? (Y/N): ')
  250 FORMAT(/,'$DO YOU WANT TO USE THE DEBUGGER? (Y/N): ')
      END
      SUBROUTINE INPUT3
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALLED BY INIT.
C
C IN FILE IO5281.FOR
C CODED BY D. KOSOWER, MARCH, 1979.
C RECODED BY M.SPIVAKOVSKY, JUNE 1980.
C
C ALLOWS SELECTIVE INPUT, SELECTIVE CORRECTIONS, ETC.
C
$INCLUDE 9,IO6.CMN
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,BLONG.CMN
```

```
C
      COMMON /INPUTS/ KR, KO, KV, KW,
     1                ITEM(25), VAL1(25), VAL2(25), INCNT
C
      LOGICAL CIRCLE,XOUT,YOUT,VOUT
      PARAMETER (PI = 3.1415927D0)
C
      KO = 0
      IBATCH = 0
C
C GET THE NUMBER OF ROOMS IN THE RUN.
  111 WRITE(IWTTY,112) NROOMS
  112 FORMAT(X,'ENTER THE NUMBER OF ROOMS IN THE RUN. DEFAULT IS ',I2)
      READ(IRTTY,113,ERR=111) NR1
  113 FORMAT(I2)
      IF (NR1 .GT. 0 .AND. NR1.LE. MAXRMS) NROOMS=NR1
      NWALLS = NROOMS
      DO 114 KW=1,NWALLS
        WLINK(1,KW) = KW
        WLINK(2,KW) = 0
  114 CONTINUE
C
C
C INPUT BY ROOM (OBJECTS ARE NESTED)
      DO 450 KR=1,NROOMS
        IF (IBATCH .EQ. 0) THEN
          WRITE (IWTTY,780)
          WRITE (IWTTY,770)
          READ (IRTTY,760)IDUM
        END IF
   10   CALL DISP(1,*80)
   20   INCNT=INCNT+1
        IF (ITEM(INCNT) .EQ. 0) GO TO 10
        V1 = VAL1(INCNT)
        GO TO (30,40,50,60,70),ITEM(INCNT)
        CALL COPINP(1)
        GOTO 20
C INPUT DIMENSIONS
   30   ZLRZX(KR) = V1
        GO TO 20
   40   ZLRZY(KR) = V1
        GO TO 20
   50   ZHRZZ(KR) = V1
        GO TO 20
   60   IV1 = INT(V1)
        IF (IV1 .GT. MAXOBS  .OR.  IV1 .LT. 0) THEN
          WRITE (IWTTY,790)
          GOTO 10
        END IF
        OBJS(KR) = IV1
        GO TO 20
   70   ZKAZZ(KR) = V1
        GO TO 20
C
C INPUT BY OBJECTS IN THE ROOM.
   80   DO 440 KO = 1,OBJS(KR)
C
C THE OBJECT IS ASSUMED TO BE CIRCULAR UNLESS THE USER
C SPECIFIES ITS SURFACE AREA BY GIVING RECTANGULAR
C DIMENSIONS
          CIRCLE = .TRUE.
C INPUT GEOMETRIC PARAMETERS FOR OBJECT
   90     CALL DISP(2,*210)
  100     INCNT=INCNT+1
          IF (ITEM(INCNT) .EQ. 0) GO TO 90
          V1 = VAL1(INCNT)
          GO TO (110,120,130,140,150,160,170,190,180,200),
     1     ITEM(INCNT)
          CALL COPINP(2)
          GOTO 100
  110     ZXOZZ(KO,KR) = V1
          GO TO 100
  120     ZYOZZ(KO,KR) = V1
          GO TO 100
  130     ZHOZZ(KO,KR) = V1
          GO TO 100
  140     ANGH(KO,KR) = V1
          GO TO 100
  150     ANGV(KO,KR) = V1
          GO TO 100
  160     ZNOZZ(KO,KR) = V1
          GO TO 100
  170     ZRFZO(KO,KR) = V1
          GO TO 100
  180     ZRFZM(KO,KR) = V1
          GO TO 100
  190     ZRFZI(KO,KR) = V1
C OBJECT IS CIRCULAR
          CIRCLE = .TRUE.
          GO TO 100
  200     V2 = VAL2(INCNT)
          ZRFZI(KO,KR) = SQRT(V1*V2/PI)
          ZRFZM(KO,KR) = SQRT((V1*V2+ZNOZZ(KO,KR)*2.D0*(V1+V2))/PI)
C OBJECT IS RECTANGULAR
          CIRCLE = .FALSE.
          XEDGE = V1/2.D0
          YEDGE = V2/2.D0
          GO TO 100
  210     IF (CIRCLE) THEN
            XEDGE = ZRFZI(KO,KR)
            YEDGE = ZRFZI(KO,KR)
          END IF
C       XLOBND = XEDGE
C       XHIBND = ZLRZX(KR)-XEDGE
C       YLOBND = YEDGE
C       YHIBND = ZLRZY(KR)-YEDGE
C       VHIBND = ZHRZZ(KR)
C       VLOBND = 0.
CC CHECK TO MAKE SURE THE OBJECT IS INSIDE THE ROOM
C 9000  XOUT = ZXOZZ(KO,KR).LT.XLOBND .OR. ZXOZZ(KO,KR).GT.XHIBND
C       YOUT = ZYOZZ(KO,KR).LT. YLOBND .OR. ZYOZZ(KO,KR).GT.YHIBND
C       VOUT = ZHOZZ(KO,KR).LT.VLOBND .OR. ZHOZZ(KO,KR).GT.VHIBND
C       IF (XOUT .OR. YOUT .OR. VOUT) THEN
C        WRITE (IWTTY,900)
C 9100   IF(XOUT) THEN
C         WRITE(IWTTY,910) 'X-COORDINATE',XLOBND,XHIBND
C         READ (IRTTY,920,ERR=9100) ZXOZZ(KO,KR)
C         XOUT = ZXOZZ(KO,KR).LT.XLOBND .OR. ZXOZZ(KO,KR).GT.XHIBND
C         GO TO 9100
C        END IF
C 9200   IF (YOUT) THEN
C         WRITE (IWTTY,910) 'Y-COORDINATE',YLOBND,YHIBND
C         READ (IRTTY,920,ERR=9200) ZYOZZ(KO,KR)
C         YOUT = ZYOZZ(KO,KR).LT. YLOBND .OR. ZYOZZ(KO,KR).GT.YHIBND
C         GO TO 9200
C        END IF
```

```
C 9300  IF (VOUT) THEN
C         WRITE (IWTTY,910) 'HEIGHT',VLOBND,VHIBND
C         READ (IRTTY,920,ERR=9300) ZHOZZ(KO,KR)
C         VOUT = ZHOZZ(KO,KR).LT.VLOBND .OR. ZHOZZ(KO,KR).GT.VHIBND
C         GO TO 9300
C       END IF
C       GO TO 9000
C      END IF
C INPUT PHYSICAL PARAMETERS FOR OBJECT
          CALL DISP(6,*430)
  220     INCNT=INCNT+1
          IF (ITEM(INCNT) .EQ. 0) GO TO 210
          V1 = VAL1(INCNT)
  230     GO TO (240,250,260,270,280,290,300,
     1       310,320,330,340,350,360,370,380,
     2       390,400,410,420),ITEM(INCNT)
          CALL COPINP(5)
          GOTO 220
  240     ZJOZZ(KO,KR) = V1
          GO TO 220
  250     ZCOZZ(KO,KR) = V1
          GO TO 220
  260     VMOZZ(KO,KR) = V1
          GO TO 220
  270     EB(KO,KR) = V1
          GO TO 220
  280     XGAMMA(KO,KR) = V1
          GO TO 220
  290     XGAMAS(KO,KR) = V1
          GO TO 220
  300     CHI(KO,KR) = V1
          GO TO 220
  310     QF(KO,KR) = V1
          GO TO 220
  320     QVAP(KO,KR) = V1
          GO TO 220
  330     ZMOZO(KO,KR) = V1
          GO TO 220
  340     ZKCIG(KO,KR) = V1
          GO TO 220
  350     ZKCPY(KO,KR) = V1
          GO TO 220
  360     FCO2(KO,KR) = V1
          GO TO 220
  370     FCO(KO,KR) = V1
          GO TO 220
  380     FS(KO,KR) = V1
          GO TO 220
  390     FH2O(KO,KR) = V1
          GO TO 220
  400     PSIO(KO,KR) = V1
          GOTO 220
  410     AFIRE(KO,KR) = V1
          GOTO 220
  420     TMFGZ(KO,KR) = V1
          GO TO 220
  430     BETA(KO,KR) = SQRT(ZGOZZ(KO,KR)/PI)/ZJOZZ(KO,KR)
          ZGOZZ(KO,KR) = ZJOZZ(KO,KR) / (VMOZZ(KO,KR)*ZCOZZ(KO,KR))
  440   CONTINUE
  450 CONTINUE
C
C
C NOW INPUT VENT PARAMETERS.
```

```
  460 WRITE (IWTTY,800)
      READ (IRTTY,760,ERR=460) NVENTS
      IF (NVENTS .LT. 0 .OR. NVENTS .GT. MAXVTS) THEN
        WRITE (IWTTY,790)
        GO TO 460
      ELSE IF (NVENTS .EQ. 0) THEN
        NVENTS = 1
      END IF
C
      DO 550 KV=1,NVENTS
  470   CALL DISP(3,*550)
  480   INCNT=INCNT+1
        IF (ITEM(INCNT) .EQ. 0) GO TO 470
        V1 = VAL1(INCNT)
        IV1 = INT(V1)
  490   GO TO (500,510,520,530,540),ITEM(INCNT)
        CALL COPINP(3)
        GOTO 480
  500   ZBVZZ(KV) = V1
        GO TO 480
  510   ZHVZZ(KV) = V1
        GO TO 480
  520   ZHTZZ(KV) = V1
        GO TO 480
  530   VLINK(1,KV) = IV1
        GO TO 480
  540   VLINK(2,KV) = IV1
        GO TO 480
  550 CONTINUE
C
C
C DATA FOR WALLS.
      DO 640 KW = 1,NWALLS
  560   CALL DISP(4,*630)
  570   INCNT=INCNT+1
        IF (ITEM(INCNT) .EQ. 0) GO TO 560
        V1 = VAL1(INCNT)
  580   GO TO (590,600,610,620),ITEM(INCNT)
        CALL COPINP(4)
        GOTO 570
  590   ZNWZZ(KW) = V1
        GO TO 570
  600   ZJWZZ(KW) = V1
        GO TO 570
  610   ZCWZZ(KW) = V1
        GO TO 570
  620   VMWZZ(KW) = V1
        GO TO 570
  630   ZGWZZ(KW) = ZJWZZ(KW) / (VMWZZ(KW)*ZCWZZ(KW))
  640 CONTINUE
C
C
C INPUT OF THINGS THAT AREN'T SUBSCRIPTED.
  650 CALL DISP(5,*750)
  660 INCNT=INCNT+1
      IF (ITEM(INCNT) .EQ. 0) GO TO 650
      V1 = VAL1(INCNT)
      GO TO (670,680,690,700,710,720,730,740), ITEM(INCNT)
  670 CP = V1
      GO TO 660
  680 ZLAMDA = 1.D0/V1
      GO TO 660
  690 ZKAOUT = V1
```

```
      GO TO 660
  700 ZOWZM = V1
      GO TO 660
  710 ZOWZN = V1
      GO TO 660
  720 ZOAZZ = V1
      GO TO 660
  730 ALPHA = V1
      GO TO 660
  740 CD = V1
      GO TO 660
  750 VMAOUT = 353.100 / ZKAOUT
      RETURN
C
  760 FORMAT (I1)
  770 FORMAT ('0 HIT <RET> TO CONTINUE.')
  780 FORMAT (/,T4,'WHEN ENTERING THE ROOM GEOMETRY BELOW, ',
     1            'NOTE THAT THE CARTESIAN COORDINATE'/,
     2            ' SYSTEM WE USE IS: FACING THE FRONT OF THE ',
     3            'BUILDING, THE ORIGIN IS AT THE BOTTOM'/,
     4            ' OF THE GROUND FLOOR, EXTREME LEFT-HAND SIDE,',
     5            ' FRONT.  FOR A SINGLE ROOM WITH A'/,
     6            ' SINGLE DOOR, "THE FRONT" IS DEFINED AS THE WALL',
     7            ' WITH THE DOOR IN IT.  IF THERE'/' IS MORE THAN',
     8            ' ONE DOOR, THE DEFINITION IS UP TO THE USER.'/,
     9        T4,'THE X-, Y-, AND Z- COORDINATES GIVE DISTANCES',
     1  ' TO THE RIGHT, REAR, AND TOP'/' OF THE ROOM,'
     2            ' THUS CONSTITUTING A STANDARD RIGHT-HANDED',
     3  ' COORDINATE SYSTEM.')
  790 FORMAT (' NUMBER TOO LARGE OR NEGATIVE')
  800 FORMAT (24(/),'$DESIRED NUMBER OF VENTS ',
     1 '(DEFAULT = 1, MAX = 5): ')
  810 FORMAT(' THE OBJECT IS NOW POSITIONED OUTSIDE OF THE ROOM.')
  820 FORMAT(' ENTER A NEW ',A,' FOR THE OBJECT IN ',/,
     1      '$THE RANGE [',1PE14.7,',',1PE14.7,'] (E FORMAT): ')
  830 FORMAT(E14.7)
      END
      SUBROUTINE RECAP(IDEVIC)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALLED BY INIT. IN FILE IO5281.FOR.
C CODED BY JOSH GAHM, FEB. 1981.
C MODIFIED BY MATT RABUZZI, MAR. 1981.
C
C THIS SUBROUTINE OUTPUTS A RECAPITULATION OF THE INPUT DATA TO
C DEVICE IDEVIC.  IF IDEVIC = IWTTY, OUTPUT IS BROKEN UP INTO TABLES
C TO FIT ON THE TERMINAL SCREEN.
C
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,BLDNG.CMN
$INCLUDE 9,IO6.CMN
C
      CHARACTER*1 IDUM
C
C OUTPUT PARAMETERS BY ROOMS
C
      DO 10 KR = 1,NROOMS
        WRITE (IDEVIC,60) KR,ZLRZX(KR),ZLRZY(KR),ZHRZZ(KR),ZKAZZ(KR)
        IF (IDEVIC .EQ. IWTTY) THEN
          WRITE (IWTTY,40)
          READ (IRTTY,50) IDUM
        END IF
C
```

```
C OUTPUT BY OBJECTS WITHIN THAT ROOM
C
        DO 10 KO=1,OBJS(KR)
          WRITE (IDEVIC,70) KO,KO,ZXOZZ(KO,KR),ZYOZZ(KO,KR),ZHOZZ(KO,KR)
     1,                    ANGH(KO,KR),ANGV(KO,KR),
     2                   ZNOZZ(KO,KR),VMOZZ(KO,KR),
     3                   ZMOZO(KO,KR), ZRFZO(KO,KR), ZRFZM(KO,KR),
     4                   ZRFZI(KO,KR),
     5                   ZCOZZ(KO,KR),ZJOZZ(KO,KR),EB(KO,KR),CHI(KO,KR)
          WRITE (IDEVIC,80) QF(KO,KR),QVAP(KO,KR),
     1                   ZKOPY(KO,KR), ZKOIG(KO,KR), XGAMMA(KO,KR),
     2                    XGAMAS(KO,KR),
     3                   FCO2(KO,KR),FCO(KO,KR),FS(KO,KR),FH2O(KO,KR),
     4                AFIRE(KO,KR)
          IF (FIRTYP(KO,KR) .EQ. 3) WRITE (IDEVIC,90) TMFGZ(KO,KR)
          IF (IDEVIC .EQ. IWTTY) THEN
            WRITE (IWTTY,40)
            READ (IRTTY,50) IDUM
          END IF
   10 CONTINUE
C
C OUTPUT BY VENTS
C
      DO 20 KV=1,NVENTS
   20 WRITE (IDEVIC,100) KV,ZBVZZ(KV),ZHVZZ(KV),ZHTZZ(KV)
      IF (IDEVIC .EQ. IWTTY) THEN
        WRITE (IWTTY,40)
        READ (IRTTY,50) IDUM
      END IF
C
C OUTPUT BY WALLS
C
      DO 30 KW = 1, NWALLS
   30 WRITE (IDEVIC,110) KW,ZNWZZ(KW),VMWZZ(KW),ZCWZZ(KW),ZJWZZ(KW)
C
C OUTPUT PARAMETERS WHICH ARE NOT INDEXED
C
      WRITE (IDEVIC,120) CP,1.D0/ZLAMDA,ZOAZZ,ALPHA,ZOWZM,ZOWZN,CD
      RETURN
C
C
   40 FORMAT ('0TYPE <RET> TO CONTINUE')
   50 FORMAT(A1)
   60 FORMAT (T3,'ROOM NUMBER',I2,':',/,
     1        T7,'DIMENSIONS(X,Y,Z)  =      (',3(F12.4),')',/,
     2        T7,'AMBIENT TEMPERATURE=',3X,F6.1)
   70 FORMAT (T5,'OBJECT NUMBER',I2,' (ID=',I2,') :',/,
     1        T7,'COORDINATES(X,Y,Z) =      (',3(F12.4),')',/,
     2        T7,'ANGLE WITH HORIZONTAL=',2X,F6.2,
     3        T40,'ANGLE WITH XZ-PLANE=',10X,F6.2,/,
     4        T7,'THICKNESS=',13X,F9.4,T40,'DENSITY=',22X,F6.2,/,
     5 T7,'INITIAL MASS=',10X,F9.4,T40,'INITIAL RADIUS=',14X,F9.4,/,
     6 T7,'MAXIMUM RADIUS=',8X,F9.4,T40,'OBJECT RADIUS=',15X,F9.4,/,
     7 T7,'SPECIFIC HEAT=',8X,F6.0,T40,'THERMAL CONDUCTIVITY=',
     8        10X,F7.4,/,T7,'EMISSIVITY=',14X,F5.2,
     9        T40,'CHI(FRACTION OF HEAT RELEASED)=',F5.2)
   80 FORMAT (T7,'HEAT OF COMBUSTION=',2X,1PE10.3,
     1        T40,'HEAT OF VAPORIZATION=',6X,1PE10.3,/,T7,
     2'PYROLIZATION TEMP=',5X,0PF6.1,T40,'IGNITION TEMP=',15X,F6.1,/,
     3        T7,'AIR/FUEL MASS RATIO=',4X,F6.2,
     4        T40,'STOICHIOMETRIC MASS RATIO=',4X,F6.2,/,
     5        T7,'FCO2(CO2 MASS/FUEL MASS)=',F6.3,T40,'FCO(CO MASS',
     6        '/FUEL MASS)=',8X,F6.3,/,T7,'FS(SMOKE MASS/FUEL MASS)=',
     7          F6.3,T40,'FH2O(H2O MASS/FUEL MASS)=',6X,F6.3,/,T7,
     8          'A(FIRE SPREAD PARAMETER)=',F6.4)
   90 FORMAT (T7,'GAS FLOW RATE OF THE GAS BURNER (IN KG/SEC) = ',
     1          F6.3)
  100 FORMAT (T3,'VENT NUMBER',I2,':',/,
     1          T7,'(WIDTH, HEIGHT, DEPTH) = (',3(F12.4),')')
  110 FORMAT (T3,'WALL NUMBER',I2,':',/,T7,'THICKNESS=',9X,F9.4,T40,
     1          'DENSITY=',22X,F6.2,/,T7,'SPECIFIC HEAT=',8X,F6.0,
     2          T40,'THERMAL CONDUCTIVITY=',10X,F7.4)
  120 FORMAT (T3,'PHYSICAL CONSTANTS:',/,T7,'SPECIFIC HEAT OF AIR=',X,
     1          F6.0,T40,'ABSORPTION COEFF OF FLAME=',5X,F5.2,/,
     2          T5,'FOR AIR:',/,T7,'HEAT TRANSFER COEFF=',4X,F6.2,T40,
     3          'PLUME ENTRAINMENT COEFF=',7X,F5.2,/,
     4          T5,'FOR LAYER GASES:'/,T7,'MAX. HEAT TRANSFER COEFF=',F6.2
     5          T40,'MIN. HEAT TRANSFER COEFF=',6X,F5.2,/,
     6          T5,'FOR VENTS:',/,T7,'FLOW COEFFICIENT=',8X,F5.2,//)
      END
      SUBROUTINE SELSUB
      IMPLICIT REAL*8 (A-H,O-Z)
C
C IN FILE IO5281.FOR.
C CODED BY J. GAHM, FEBRUARY 1981.
C MODIFIED BY M.RABUZZI, 3-10-81.
C
C CALLED BY INIT
C
C ALLOWS THE USER TO SELECT WHICH VERSION OF EACH PHYSICS SUBROUTINE
C HE WISHES TO USE.
C
C (2/20/81) AT PRESENT, THE ONLY SUBROUTINES WHICH ARE AVAILABLE IN MULT
C VERSIONS ARE TMPO AND ABSRB.
C
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,BLDNG.CMN
$INCLUDE 9,IO6.CMN
C
      CHARACTER*1 ICHAR
C
C ASK USER QUESTIONS
C
   10 WRITE (IWTTY,50)
      READ (IRTTY,60) ICHAR
      IF (ICHAR .EQ. 'N' .OR. ICHAR .EQ. ' ') GO TO 40
      IF (ICHAR .NE. 'Y') GO TO 10
C
C INPUT NON-DEFAULT VERSION NUMBERS
C
   20 WRITE (IWTTY,80) 'TMPO ','(01 OR 02)','TMPO02'
      READ (IRTTY,70,ERR=20) VERSUN(1)
      IF (VERSUN(1) .LE. 0) THEN
        VERSUN(1)=2
      ELSE IF (VERSUN(1) .GT. 2) THEN
        GOTO 20
      END IF
   30 WRITE (IWTTY,80) 'ABSRB','(1, 2, OR 3)','ABSRB2'
      READ (IRTTY,70,ERR=30) VERSUN(2)
      IF (VERSUN(2) .LE. 0) THEN
        VERSUN(2)=2
      ELSE IF (VERSUN(2) .GT. 3) THEN
        GOTO 30
      END IF
C
C LIST SUBROUTINES IN USE ON DISK FILE
```

```
C
   40 RETURN
C
   50 FORMAT (/,'$CHANGE FROM STANDARD SET OF PHYSICS ',
     1        'SUBROUTINES? (Y/N): ')
   60 FORMAT(A1)
   70 FORMAT(I2)
   80 FORMAT('OWHICH VERSION OF ',A6,' DO YOU WANT ',A15,/,
     1        '$DEFAULT IS ',A6,': ')
   90 FORMAT ('OPHYSICS SUBROUTINES IN USE (ONLY FOR',
     1        ' SUBROUTINES WITH MULTIPLE VERSIONS):')
  100 FORMAT (' ',A6,', VERSION ',I3)
      END
      BLOCK DATA SR
      IMPLICIT INTEGER (A-Z)
C
C THIS DATA BLOCK INITIALIZES THE VARIABLES IN THE INTERFACE
C COMMON BLOCKS.
C
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,BLDNG.CMN
      INTEGER RMSM1, SIDEM2, VSDSM2
      PARAMETER (RMSM1 = MAXRMS - 1, SIDEM2 = 2*MAXWLS - 2,
     1      VSDSM2 = 2*MAXVTS - 2)
      INTEGER OBJSM0,OBJSM1,OBJSM2
      PARAMETER (OBJSM0 = MAXOBS*MAXRMS,
     1           OBJSM1 = OBJSM0 - 1,
     2           OBJSM2 = OBJSM0 - 2)
C
C VARIABLES IN COMMON /BLDNG/:
      DATA NRCOMS,NWALLS,NVENTS /3*1/, OBJS /2,RMSM1*0/,
     1      WLINK /1,0,SIDEM2*0/, VLINK /1,0,VSDSM2*0/
      DATA ZKAOUT,VMAOUT /300.D0,1.177D0/
      DATA CP/1004.D0/
      DATA VERSUN /2,2,3*1/
      DATA STATE/FLAME,OBJSM1*HEATNG/,ZXOZZ/.84D0,2.08D0,OBJSM2*0.D0/,
     1ZYOZZ/2*2.818D0,OBJSM2*0.D0/,ZHOZZ/.61D0,.864D0,OBJSM2*.61D0/,
     2      ZNOZZ/OBJSM0*.1D0/, ZJOZZ/OBJSM0*0.054D0/,
     3      ZGOZZ/OBJSM0*5.92D-7/, ZCOZZ/OBJSM0*1900.D0/,
     4      VMOZZ/OBJSM0*48.0D0/, EB/OBJSM0*.98D0/, CHI/OBJSM0*.65D0/,
     5      ZMOZ0/6.852D0,OBJSM1*1.0963D0/, ZRFZ0/OBJSM0*.037D0/,
     6      ZRFZM/.9677D0,OBJSM1*.4657D0/, QF/OBJSM0*28.7D6/,
     7      QVAP/OBJSM0*2.054D6/, ZOAZZ/10.0D0/, ALPHA/0.1D0/,
     8      ZTIG/0.D0,OBJSM1*-1.D0/, ZKOIG/727.D0,OBJSM1*740.D0/,
     9      ZKCPY/OBJSM0*600.D0/, XGAMMA/OBJSM0*14.45D0/
      DATA XGAMAS/OBJSM0*9.85D0/, ZRFZI/.8598D0,OBJSM1*.3439D0/,
     1      FCO2/OBJSM0*1.504D0/, FCO/OBJSM0*.0133D0/,
     2      FS/OBJSM0*.241D0/, FH2O/OBJSM0*.7135D0/, ZKFZZ/1260.D0/,
     3      ZLAMDA/.6452D0/, TMFGZ/OBJSM0*.01D0/,ZUFZZ/1.55D0/,
     4      FIRTYP/GROWNG,OBJSM1*0/, ANGV/OBJSM0*0.D0/,
     5      ANGH/OBJSM0*0.D0/, AFIRE/OBJSM0*.0109D0/,
     6      PSIO/OBJSM0*30.D0/, TPSIO/OBJSM0*0.D0/
      DATA ZRFZZP/3.7D-2,OBJSM1*0.D0/,TMOZZP/OBJSM0*0.D0/,
     1      ZHFZZP/OBJSM0*0.D0/, ZMOZZP/6.852D0,OBJSM1*1.0963D0/,
     2      ZKOZZP/OBJSM0*300.D0/
      DATA ZLRZY/MAXRMS*3.6576D0/, ZLRZX/MAXRMS*2.4384D0/,
     1      ZHRZZ/MAXRMS*2.4384D0/, ZKAZZ/MAXRMS*300.D0/,
     2      VMAZZ/MAXRMS*1.177D0/, FIROOM/.TRUE.,RMSM1*.FALSE./
      DATA ZBVZZ/MAXVTS*.7620D0/, ZHVZZ/MAXVTS*2.0320D0/,
     1      ZHTZZ/MAXVTS*.4064D0/, CD/.68D0/
      DATA ZNWZZ/MAXWLS*.0254D0/, ZJWZZ/MAXWLS*.134D0/,
     1      ZGWZZ/MAXWLS*1.577D-7/, ZCWZZ/MAXWLS*1062.D0/,
     2      VMWZZ/MAXWLS*800.D0/, ZOWZM/50.D0/, ZOWZN/5.D0/
       END
```

PART 2 - DBLES

```
      SUBROUTINE RECAP(IDEVIC)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CODED BY JOSH GAHM, FEB. 1981.
C MODIFIED BY MATT RABUZZI, MAR. 1981.
C
C THIS SUBROUTINE OUTPUTS A RECAPITULATION OF THE INPUT DATA TO
C DEVICE IDEVIC.  IF IDEVIC = IWTTY, OUTPUT IS BROKEN UP INTO TABLES
C TO FIT ON THE TERMINAL SCREEN.
C
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,BLDNG.CMN
$INCLUDE 9,IO6.CMN
C
      CHARACTER*1 IDUM
C
C OUTPUT PARAMETERS BY ROOMS
C
      DO 10 KR = 1,NROOMS
        WRITE (IDEVIC,60) KR,ZLRZX(KR),ZLRZY(KR),ZHRZZ(KR),ZKAZZ(KR)
        IF (IDEVIC .EQ. IWTTY) THEN
          WRITE (IWTTY,40)
          READ (IRTTY,50) IDUM
        END IF
C
C OUTPUT BY OBJECTS WITHIN THAT ROOM
C
        DO 10 KO=1,OBJS(KR)
          WRITE (IDEVIC,70) KO,KO,ZXOZZ(KO,KR),ZYOZZ(KO,KR),ZHOZZ(KO,KR)
     1,                         ANGH(KO,KR),ANGV(KO,KR),
     2                         ZNOZZ(KO,KR),VMOZZ(KO,KR),
     3                         ZMOZO(KO,KR), ZRFZO(KO,KR), ZRFZM(KO,KR),
     4                         ZRFZI(KO,KR),
     5                         ZCOZZ(KO,KR),ZJOZZ(KO,KR),EB(KO,KR),CHI(KO,KR)
          WRITE (IDEVIC,80) QF(KO,KR),QVAP(KO,KR),
     1                         ZKOPY(KO,KR), ZKOIG(KO,KR), XGAMMA(KO,KR),
     2                          XGAMAS(KO,KR),
     3                         FCO2(KO,KR),FCO(KO,KR),FS(KO,KR),FH2O(KO,KR),
     4                      AFIRE(KO,KR)
          IF (FIRTYP(KO,KR) .EQ. 3) WRITE (IDEVIC,90) TMFGZ(KO,KR)
          IF (IDEVIC .EQ. IWTTY) THEN
            WRITE (IWTTY,40)
            READ (IRTTY,50) IDUM
          END IF
   10 CONTINUE
C
C OUTPUT BY VENTS
C
      DO 20 KV=1,NVENTS
   20 WRITE (IDEVIC,100) KV,ZBVZZ(KV),ZHVZZ(KV),ZHTZZ(KV)
      IF (IDEVIC .EQ. IWTTY) THEN
        WRITE (IWTTY,40)
        READ (IRTTY,50) IDUM
      END IF
C
C OUTPUT BY WALLS
C
      DO 30 KW = 1, NWALLS
   30 WRITE (IDEVIC,110) KW,ZNWZZ(KW),VMWZZ(KW),ZCWZZ(KW),ZJWZZ(KW)
C
C OUTPUT PARAMETERS WHICH ARE NOT INDEXED
C
```

```
      WRITE (IDEVIC,120) CP,1.D0/ZLAMDA,ZOAZZ,ALPHA,ZOWZM,ZOWZN,CD
      RETURN
C
C
   40 FORMAT ('0TYPE <RET> TO CONTINUE')
   50 FORMAT(A1)
   60 FORMAT (T3,'ROOM NUMBER',I2,':',/,
     1        T7,'DIMENSIONS(X,Y,Z)  =      (',3(F12.4),')',/,
     2        T7,'AMBIENT TEMPERATURE=',3X,F6.1)
   70 FORMAT (T5,'OBJECT NUMBER',I2,' (ID=',I2,') :',/,
     1        T7,'COORDINATES(X,Y,Z) =      (',3(F12.4),')',/,
     2        T7,'ANGLE WITH HORIZONTAL=',2X,F6.2,
     3        T40,'ANGLE WITH XZ-PLANE=',10X,F6.2,/,
     4        T7,'THICKNESS=',13X,F9.4,T40,'DENSITY=',22X,F6.2,/,
     5 T7,'INITIAL MASS=',10X,F9.4,T40,'INITIAL RADIUS=',14X,F9.4,/,
     6 T7,'MAXIMUM RADIUS=',8X,F9.4,T40,'OBJECT RADIUS=',15X,F9.4,/,
     7 T7,'SPECIFIC HEAT=',8X,F6.0,T40,'THERMAL CONDUCTIVITY=',
     8        10X,F7.4,/,T7,'EMISSIVITY=',14X,F5.2,
     9        T40,'CHI(FRACTION OF HEAT RELEASED)=',F5.2)
   80 FORMAT (T7,'HEAT OF COMBUSTION=',2X,1PE10.3,
     1        T40,'HEAT OF VAPORIZATION=',6X,1PE10.3,/,T7,
     2'PYROLIZATION TEMP=',5X,0PF6.1,T40,'IGNITION TEMP=',15X,F6.1,/,
     3        T7,'AIR/FUEL MASS RATIO=',4X,F6.2,
     4        T40,'STOICHIOMETRIC MASS RATIO=',4X,F6.2,/,
     5        T7,'FCO2(CO2 MASS/FUEL MASS)=',F6.3,T40,'FCO(CO MASS',
     6        '/FUEL MASS)=',8X,F6.3,/,T7,'FS(SMOKE MASS/FUEL MASS)=',
     7        F6.3,T40,'FH2O(H2O MASS/FUEL MASS)=',6X,F6.3,/,T7,
     8        'A(FIRE SPREAD PARAMETER)=',F6.4)
   90 FORMAT (T7,'GAS FLOW RATE OF THE GAS BURNER (IN KG/SEC) = ',
     1        F6.3)
  100 FORMAT (T3,'VENT NUMBER',I2,':',/,
     1        T7,'(WIDTH, HEIGHT, DEPTH) = (',3(F12.4),')')
  110 FORMAT (T3,'WALL NUMBER',I2,':',/,T7,'THICKNESS=',9X,F9.4,T40,
     1        'DENSITY=',22X,F6.2,/,T7,'SPECIFIC HEAT=',8X,F6.0,
     2        T40,'THERMAL CONDUCTIVITY=',10X,F7.4)
  120 FORMAT (T3,'PHYSICAL CONSTANTS:',/,T7,'SPECIFIC HEAT OF AIR=',X,
     1        F6.0,T40,'ABSORPTION COEFF OF FLAME=',5X,F5.2,/,
     2        T5,'FOR AIR:',/,T7,'HEAT TRANSFER COEFF=',4X,F6.2,T40,
     3        'PLUME ENTRAINMENT COEFF=',7X,F5.2,/,
     4        T5,'FOR LAYER GASES:'/,T7,'MAX. HEAT TRANSFER COEFF=',F6.2
     5        T40,'MIN. HEAT TRANSFER COEFF=',6X,F5.2,/,
     6        T5,'FOR VENTS:',/,T7,'FLOW COEFFICIENT=',8X,F5.2,//)
      END
      SUBROUTINE ALDIFF(X,XMAX,TIME,H,HSTORE,HMIN,EPS,
     1  RESULT,ORDER,MAXVRS,MAXODR,
     1  OLDX,ERROR,ZERO,Z,JAC,PIVCT)
      IMPLICIT INTEGER (A-Z)
      INTEGER RESULT,ORDER,MAXVRS,FNUM,GNUM,HNUM,INUM,PIVCT(*),MAXODR
      REAL*8 X(*),XMAX(*),TIME,H,HMIN,HMAX,EPS,OLDX(*),
     1  ERROR(*),ZERO(*),Z(*),JAC(MAXVRS,*),HSTORE
C
C FIRST ORDER ROUTINE FOR SOLVING A SYSTEM OF
C ALGEBRAIC-DIFFERENTIAL EQUATIONS. THE ROUTINE IS BASED ON GEAR'S
C DIFSUB. THE FORM OF THE EQUATIONS SOLVED IS
C  O = F(U,V,W,T)
C  V'= G(U,V,W,T)
C  W = H(U,V,W,T)
C              Y = I(U,V,W,T)
C  WHERE U, V, W, AND Y HAVE, RESPECTIVELY, THE DIMENSIONS OF F, G, H,
C  AND I.
C THE PROGRAM REQUIRES A ROUTINE NAMED
C  FUNCT(X,Z,T)
C THE INPUT TO THE ROUTINE IS X AND T. X IS AN ARRAY CONTAINING
```

```
C U,V,W, AND Y STORED CONTIGUOUSLY AND FUNCT RETURNS THE VALUES
C FOR F,G,H, AND I STORED IN Z. FNUM, GNUM, HNUM, AND INUM ARE
C THE DIMENSIONS OF U,V,W,Y RESPECTIVELY.
C THIS VERSION IS A HYBRID OF THE NEWTON AND GAUSS-SEIDEL
C PACKAGES.  IT RUNS IN GAUSS SEIDEL UNTIL THE CORRECTOR
C FAILS TO CONVERGE AT THE MINIMUM TIME STEP AND THEN
C SWITCHES OVER TO NEWTON.  IN SUBSEQUENT STEPS, EVERY TIME
C THE JACOBIAN IS REEVALUATED THE PROGRAM FIRST TRIES TO
C CONVERGE THE CORRECTOR USING GAUSS-SEIDEL.  AS SOON AS
C ONE OF THESE ATTEMPTS SUCCEEDS, THE PROGRAM SWITHCES BACK
C TO GAUSS-SEIDEL.  OTHERWISE, IT EVALUATES THE FULL JACOBIAN
C AND REMAINS IN NEWTON MODE.
C
C
C INPUT:
C     X -- THE LATEST SET OF CONVERGED VALUES FOR THE VARIABLES
C          AND THEIR SCALED DERIVATIVES.
C     XMAX -- ARRAY CONTAINING A SIGNIFICANT VALUE FOR EACH VARIABLE.
C     TIME -- THE TIME AT WHICH THE VARIABLES IN X ARE CONVERGED.
C     H -- THE USER'S SUGGESTED TIME STEP.  ALDIFF WILL TRY TO TAKE
C          A TIME STEP OF THIS SIZE FIRST, AND WILL CUT THE TIME STEP
C          ONLY IF IT FAILS TO CONVERGE WITH THE GIVEN TIME STEP.
C     HSTORE -- THE TIME STEP AT WHICH THE DERIVATIVES IN X ARE SCALED.
C     HMIN -- THE MINIMUM TIME STEP.
C     EPS -- THE ONE STEP ERROR TOLERANCE.
C     ORDER -- THE ORDER OF THE METHOD THAT WILL BE USED.
C     MAXVRS -- DECLARED ROW DIMENSION OF JAC.
C     ERROR --
C     JAC -- AN ESTIMATE OF THE JACOBIAN MATRIX OF THE SYSTEM OF
C            EQUATIONS DECOMPOSED INTO UPPER TRIANGULAR FORM.
C
C OUTPUT:
C     X -- THE LATEST SET OF CONVERGED VALUES FOR THE VARIABLES.
C          IF ALDIFF DOES NOT SUCCESSFULLY COMPLETE A TIME STEP,
C          X IS RESTORED TO ITS INPUT VALUES.
C     TIME -- THE TIME AT WHICH THE VARIABLES IN X ARE CONVERGED.
C     H -- A SUGGESTION FOR THE NEXT TIME STEP.  THIS IS NOT RECOM-
C          PUTED AT EVERY TIME STEP.  IF THE USER HAS SUGGESTED A
C          TIME STEP SIZE OTHER THAN THE ONE THAT ALDIFF SUGGESTS,
C          ALDIFF MAY SIMPLY RETURN THIS MODIFIED TIME STEP AS ITS
C          "SUGGESTION".
C     HSTORE -- THE STEP SIZE AT WHICH THE DERIVATIVES IN X ARE
C               SCALED.
C     RESULT -- CONTAINS A FACTOR OF -1 IF THE STEP FAILED.
C               CONTAINS A FACTOR OF 2 FOR EVERY TIME THAT
C               THE LOCAL ERROR WAS TOO LARGE.
C               CONTAINS A FACTOR OF 3 FOR EVERY TIME THAT THE
C               CORRECTOR FAILED TO CONVERGE.
C               CONTAINS A FACTOR OF 5 IF THE JACOBIAN WAS SINGULAR
C               TO MACHINE PRECISION.
C     ORDER -- A SUGGESTION FOR THE ORDER TO BE USED AT THE NEXT
C              TIME STEP.
C     OLDX -- THE VALUES THAT X CONTAINED ON INPUT.
C     ERROR -- AN ESTIMATE OF THE TRUNCATION ERROR FOR EACH VARIABLE
C              FOR THE TIME STEP.
C     ZERO -- GARBAGE.
C     Z -- GARBAGE.
C     JAC -- AN ESTIMATE OF THE JACOBIAN DECOMPOSED INTO UPPER
C            TRIANGULAR FORM.  MAY BE A DIFFERENT ESTIMATE THAN
C            ON INPUT.
C     PIVOT -- THE PIVOT VECTOR FOR THE DECOMPOSED JACOBIAN.
C
C
```

```
      INTEGER I,J,K,COUNT,SIZE,RF,RFP1,DERS,DERSP1,FIX,FIXP1
      INTEGER CALLS,CHECK,MAXCOR,METHOD,NWTN,GS,MATRIX,
     1 GSCR,NWTNCR,MAXREV
      PARAMETER (NWTN = 1, GS = 2, GSCR = 25, NWTNCR = 10,
     1 MAXREV = 30)
C EXPLANATION OF FLAGS FOR THE HYBRID:
C     METHOD -- SET TO NWTN WHEN IN NEWTON MODE AND TO GS WHEN IN
C               GAUSS-SEIDEL MODE.
C     MATRIX -- SET TO NWTN IF THE CURRENT MATRIX STORED IN JAC
C               IS A FULL JACOBIAN AND TO GS IF THE MATRIX IN JAC
C               IS ONLY A GAUSS-SEIDEL JACOBIAN.
C     CALLS -- THE NUMBER OF ATTEMPTED TIME STEPS FOR WHICH THE
C              CURRENT JACOBIAN HAS BEEN USED.
C     GSCR -- THE MAXIMUM NUMBER OF CORRECTOR ITERATIONS WHICH MAY
C             BE TAKEN WHEN JAC CONTAINS A GAUSS-SEIDEL MATRIX.
C     NWTNCR -- THE MAXIMUM NUMBER OF CORRECTOR ITERATIONS WHICH
C               MAY BE TAKEN WHEN JAC CONTAINS A FULL NEWTON MATRIX.
C     MAXREV -- THE MAXIMUM NUMBER OF ATTEMPTED TIME STEPS FOR WHICH
C               A JACOBIAN MAY BE USED WITHOUT BEING REEVALUATED.
C
      REAL*8 CTOL,OLDT,OLDH,SUM,DELTA,L(0:6,6),
     1  L20,L22,L30,L32,L33,L40,L42,L43,L44,L50,L52,L53,L54,L55,
     2 L60,L62,L63,L64,L65,L66,PR1,PR2,PR3,ENQ1,ENQ2,ENQ3,
     3  E,EUP,EDOWN,RACUM,R
      REAL*8 ERRTST(7,3)
      LOGICAL CONV
      SAVE
      PARAMETER (L20 = 2.0D0/3.0D0, L22 = 1.0D0/3.0D0,
     1 L30 = 6.0D0/11.0D0, L32 = 6.0D0/11.0D0, L33 = 1.0D0/11.0D0,
     2 L40 = 0.48D0, L42 = 0.7D0, L43 = 0.2D0, L44 = 0.02D0,
C  WE'RE NOT SURE IF L40 SHOULD BE 12/25 OR 1/2.  GEAR'S BOOK SAYS
C  BOTH.
     3 L50 = 1.20D0/2.74D0, L52 = 2.25D0/2.74D0,
     4  L53 = 8.5D0/27.4D0, L54 = 1.5D0/27.4D0, L55 = 1.0D0/274.0D0,
     5  L60 = 1.8D0/4.41D0, L62 = 5.8D0/6.3D0, L63 = 1.5D0/3.6D0,
     6  L64 = 2.5D0/25.2D0, L65 = 3.0D0/252.0D0, L66 = 1.0D0/1764.0D0)
      DATA L/1.0D0,1.0D0,0.0D0,0.0D0,0.0D0,0.0D0,0.0D0,
     1      L20,  1.0D0,L22,  0.0D0,0.0D0,0.0D0,0.0D0,
     2      L30,  1.0D0,L32,  L33,  0.0D0,0.0D0,0.0D0,
     3      L40,  1.0D0,L42,  L43,  L44,  0.0D0,0.0D0,
     4      L50,  1.0D0,L52,  L53,  L54,  L55,  0.0D0,
     5      L60,  1.0D0,L62,  L63,  L64,  L65,  L66  /
      DATA ERRTST/2.0D0,12.0D0,24.0D0,37.89D0,53.33D0,70.08D0,87.97D0,
     1    12.0D0,24.0D0,37.89D0,53.33D0,70.08D0,87.97D0,1.0D0,
     2    1.0D0,1.0D0,2.0D0,1.0D0,0.3157D0,0.07407D0,0.0139D0/
C
      DO 10 I = 1,(ORDER+1)*SIZE
        OLDX(I) = X(I)
   10 CONTINUE
      OLDH = HSTORE
      RESULT = 1
      OLDT = TIME
      RACUM = 1.0D0
      IF (H .EQ. OLDH) GOTO 60
      RACUM = H / OLDH
C  SCALE THE APPROXIMATIONS OF THE DERIVITIVES
   20 RACUM = MAX(ABS(HMIN/OLDH),RACUM)
      SUM = 1.0D0
      DO 40 J = 1,ORDER
        SUM = SUM * RACUM
        DO 30 I = 1,SIZE
          X(J*SIZE+I) = SUM * OLDX(J*SIZE+I)
   30   CONTINUE
   40 CONTINUE
      H = OLDH * RACUM
      DO 50 I = 1,SIZE
        X(I) = OLDX(I)
   50 CONTINUE
      HSTORE = H
      CHECK = ORDER + 1
C PREDICT
   60 TIME = TIME + H
      DO 90 J = 1,ORDER
        DO 80 K = J,ORDER
          COUNT = (ORDER - K + J - 1) * SIZE
          DO 70 I = 1,SIZE
            X(COUNT+I) = X(COUNT+I) + X(COUNT+SIZE+I)
   70     CONTINUE
   80   CONTINUE
   90 CONTINUE
      DO 100 I = 1,SIZE
        ERROR(I) = 0.0D0
  100 CONTINUE
C
C CORRECT.
C
C REEVALUATE THE JACOBIAN IF NECESSARY.
      IF (CALLS .GT. MAXREV) THEN
        IF (METHOD .EQ. NWTN .AND. MATRIX .EQ. GS) THEN
C EVALUATE THE FULL JACOBIAN:
          I = FIX
          MATRIX = NWTN
        ELSE
C EVALUATE THE PARTIAL DERIVATIVES ONLY FOR THE ROOT FINDING
C EQUATIONS (THIS IS AN IMPLEMENTATION OF THE GAUSS-SEIDEL
C METHOD).
          I = RF
          MATRIX = GS
        END IF
        CALL EVJAC(I,X,EPS,XMAX,TIME,RF,RFP1,DERS,H,DERSP1,
     1 L(0,ORDER),SIZE,MAXVRS,PIVOT,CONV,JAC,Z,ZERO)
C SEE IF THE MATRIX WAS SINGULAR.
        IF (CONV) THEN
          RESULT = -5 * RESULT
C PUT BACK THE LAST CONVERGED VALUES.
          TIME = OLDT
          HSTORE = OLDH
          DO 181 I=1,(ORDER+1)*SIZE
            X(I) = OLDX(I)
  181     CONTINUE
          RETURN
        END IF
        CALLS = 0
      END IF
C
      IF (MATRIX .EQ. NWTN) THEN
        I = NWTNCR
        J = FIX
        K = FIXP1
      ELSE
        I = GSCR
        J = RF
        K = RFP1
      END IF
      CALL CORECT(JAC,I,TIME,X,RFP1,DERS,SIZE,H,DERSP1,MAXVRS,
     1 J,PIVOT,K,L(0,ORDER),ERROR,RF,CTOL,XMAX,CONV,COUNT,ZERO)
```

```
      CALLS = CALLS + 1
C
      IF (.NOT. CONV) THEN
C CORRECTOR DID NOT CONVERGE
        RESULT = RESULT * 3
        IF (H .LE. HMIN*1.00001D0) THEN
C WE'RE AT THE MINIMUM TIME STEP
          IF (MATRIX .EQ. NWTN) THEN
C THE TIME STEP FAILED EVEN THOUGH THE FULL NEWTON MATRIX WAS USED.
C BOMB OUT.
            RESULT = -RESULT
C BACK UP TO THE LAST CONVERGED VALUES.
            TIME = OLDT
            HSTORE = OLDH
            DO 101 I=1,(ORDER+1)*SIZE
              X(I) = OLDX(I)
  101       CONTINUE
            RETURN
          ELSE
C WE HAVE NOT YET TRIED THE TIME STEP WITH THE FULL NEWTON MATRIX.
C SWITCH TO NEWTON'S METHOD FROM GAUSS-SEIDEL.
            METHOD = NWTN
            CALLS = MAXREV + 1
            TIME = OLDT
            H = OLDH
            GO TO 20
          END IF
        ELSE
C THE CORRECTOR DID NOT CONVERGE, BUT WE CAN STILL CUT THE TIME
C STEP AND TRY AGAIN IN GAUSS-SEIDEL.
          CALLS = MAXREV + 1
          TIME = OLDT
          H = OLDH
          RACUM = RACUM * 0.5D0
          GOTO 20
        ENDIF
      ELSE
C CORRECTOR CONVERGED
C IF WERE IN NEWTON MODE AND THE CORRECTOR JUST CONVERGED WITHOUT
C USING THE FULL NEWTON MATRIX, THEN SWITCH BACK TO GAUSS-SEIDEL
C MODE.
        IF (MATRIX.EQ.GS .AND. METHOD.EQ.NWTN) METHOD = GS
        SUM = 0.0D0
        DO 110 I = 1,SIZE
          SUM = SUM + (ERROR(I)/XMAX(I))**2
  110   CONTINUE
        IF ((H.LE.2.D0*HMIN.AND.ORDER.EQ.1) .OR. SUM .LE. E) THEN
C THE LOCAL ERROR IS WITHIN TOLERANCE
          OLDT = TIME
          NEWH = H
          DO 130 J = 2,ORDER
            DO 120 I = 1,SIZE
              X(J*SIZE+I) = X(J*SIZE+I) - L(J,ORDER) * ERROR(I)
  120       CONTINUE
  130     CONTINUE
          CHECK = CHECK - 1
          IF (CHECK .GT. 1) RETURN
          IF (CHECK .GT. 0) THEN
            DO 140 I = 1,SIZE
              ERROR(I+SIZE) = ERROR(I)
  140       CONTINUE
            RETURN
          ENDIF
```

```
    ELSE
C LOCAL ERROR TOO LARGE
     RESULT = RESULT * 2
     IF (H .LE. HMIN*1.00001D0) THEN
      RESULT = -RESULT
C RECOVER LAST CONVERGED VALUES.
      TIME = OLDT
      HSTORE = OLDH
      DO 141 I=1,(ORDER+1)*SIZE
       X(I) = OLDX(I)
 141  CONTINUE
      RETURN
     ELSE
      TIME = OLDT
      H = OLDH
      RACUM = RACUM *0.25D0
      GOTO 20
     ENDIF
    ENDIF
   ENDIF
C CALCULATE NEW STEP SIZE
   PR2 = 1.2D0 * (SUM/E) ** ENQ2
   PR3 = 1.0D32
   IF (ORDER .LT. MAXODR) THEN
    SUM = 0.0D0
    DO 150 I = 1,SIZE
     SUM = SUM + ((ERROR(I)-ERROR(I+SIZE))/XMAX(I)) ** 2
 150  CONTINUE
    PR3 = 1.4D0 * (SUM/EUP) ** ENQ3
   ENDIF
   PR1 = 1.0D32
   IF (ORDER .GT. 1) THEN
    SUM = 0.0D0
    DO 160 I = 1,SIZE
     SUM = SUM + (X(ORDER*SIZE+I)/XMAX(I)) ** 2
 160  CONTINUE
    PR1 = 1.3D0 * (SUM/EDOWN) ** ENQ1
   ENDIF
   IF (PR2 .LE. PR3) GOTO 180
   IF (PR3 .LT. PR1) GOTO 190
 170 R = 1.0D0/MAX(PR1,1.0D-4)
   CONV = .FALSE.
   ORDER = ORDER - 1
   GOTO 210
 180 IF (PR2 .GT. PR1) GOTO 170
   R = 1.0D0/MAX(PR2, 1.0D-4)
   CONV = .TRUE.
   GOTO 210
 190 R = 1.0D0/MAX(PR3, 1.0D-4)
   CONV = .FALSE.
   DO 200 I = 1,SIZE
    X((ORDER+1)*SIZE+I) = ERROR(I) * L(ORDER,ORDER) / DBLE(ORDER)
 200 CONTINUE
   ORDER = ORDER + 1
   GOTO 210
 210 CHECK = ORDER + 1
   IF (R .GT. 1.1D0 .OR. R .LT. 0.9D0) H = H * R
   IF (CONV) RETURN
C  ORDER CHANGE
   CALLS = MAXREV + 1
   ENQ1 = 0.5D0 / DBLE(ORDER)
   ENQ2 = 0.5D0 / DBLE(ORDER + 1)
   ENQ3 = 0.5D0 / DBLE(ORDER + 2)
```

```
      E = (ERRTST(ORDER,1)*EPS) ** 2
      EUP = (ERRTST(ORDER,2)*EPS) ** 2
      EDOWN = (ERRTST(ORDER,3)*EPS) ** 2
      CTOL = EPS * ENQ3 / DBLE(SIZE)
      RETURN
C
      ENTRY ADINIT(X,TIME,HSTORE,EPS,ORDER,FNUM,GNUM,HNUM,INUM)
C   ENTRY FOR INITIALIZATION
C   ADINIT IS CALLED ONCE BEFORE ANY CALL TO ALDIFF AND IT MAY BE CALLED
C   WHEN THE USER WANTS TO RESET ALDIFF TO BE A FIRST ORDER METHOD.
      METHOD = GS
      CALLS = MAXREV + 1
      RF = FNUM
      RFP1 = RF + 1
      DERS = FNUM + GNUM
      DERSP1 = DERS + 1
      FIX = DERS + HNUM
      FIXP1 = FIX + 1
      SIZE = FIX + INUM
      CALL FUNCT(X,X(SIZE+1),TIME)
      DO 220 I = 1,FNUM
        X(I+SIZE) = 0.0D0
  220 CONTINUE
      DO 230 I = RFP1,DERS
        X(I+SIZE) = X(I+SIZE) * HSTORE
  230 CONTINUE
      DO 240 I = DERSP1,SIZE
        X(I) = X(I+SIZE)
        X(I+SIZE) = 0.0D0
  240 CONTINUE
      CHECK = 2
      ORDER = 1
      ENQ1 = 0.5D0
      ENQ2 = 0.25D0
      ENQ3 = 1.0D0/6.0D0
      E = (ERRTST(1,1)*EPS) ** 2
      EUP = (ERRTST(1,2)*EPS) ** 2
      EDOWN = (ERRTST(1,3)*EPS) ** 2
      CTOL = EPS * ENQ3 / DBLE(SIZE)
      RETURN
      END
      SUBROUTINE CORECT(JAC,MAXCOR,TIME,X,FNUMP1,DERS,SIZE,
     1 H,DERSP1,MAXVRS,LEQ,PIVOT,LEQP1,L,ERROR,FNUM,CTOL,XMAX,
     2 CONV,COUNT,ZERO)
      IMPLICIT INTEGER (A-Z)
      INTEGER FNUMP1,DERS,SIZE,DERSP1,MAXVRS,LEQ,LEQP1,FNUM,
     1 COUNT,MAXCOR,PIVOT(*)
      REAL*8 JAC(MAXVRS,*),TIME,X(*),H,L,ERROR(*),CTOL,XMAX(*),
     1   ZERO(*)
      LOGICAL CONV
C
C THIS SUBROUTINE ATTEMPS TO ITERATIVELY SOLVE THE CORRECTOR EQUATION
C GIVEN AN INITIAL GUESS AND AN ESTIMATE OF THE JACOBIAN MATRIX.
C INPUT:
C      JAC -- AN ESTIMATE OF THE JACOBIAN MATRIX IN DECOMPOSED, UPPER
C             TRIANGULAR FORM.
C      MAXCOR -- THE MAXIMUM NUMBER OF ALLOWABLE ITERATIONS.
C      TIME -- THE INDEPENDENT VARIABLE OF THE SYSTEM OF EQUATIONS.
C      X -- THE INITIAL GUESS AT THE SOLUTION OF THE SYSTEM.
C      FNUMP1 -- POINTER TO THE FIRST DIFFERENTIAL EQUATION VARIABLE
C                IN THE VECTOR X.
C      DERS -- POINTER TO THE LAST DIFFERENTIAL EQUATION VARIABLE IN
C              THE VECTOR X.
```

```
C      SIZE -- NUMBER OF VARIABLES IN THE SYSTEM.
C      H -- SIZE OF THE CURRENT TIME STEP.
C      DERSP1 -- POINTER TO THE FIRST FIXED POINT VARIABLE IN THE
C                VECTOR X.
C      MAXVRS -- DECLARED ROW DIMENSION OF JAC.
C      LEQ -- NUMBER OF COLUMNS WHICH WERE EVALUATED IN THE JACOBIAN.
C      PIVOT -- PIVOT VECTOR USED BY SOLVE.
C      LEQP1 -- LEQ + 1
C      L --
C      FNUM -- POINTER TO THE LAST ROOT FINDING VARIABLE IN X.
C      CTOL -- ERROR TOLERANCE FOR THE CORRECTOR.
C      XMAX -- ARRAY CONTAINING A SIGNIFICANT VALUE FOR EACH VARIABLE
C              IN THE SYSTEM.
C      ZERO -- CONTENTS IGNORED.
C
C OUTPUT:
C      CONV -- SET TRUE IF THE CORRECTOR CONVERGED.
C      ZERO -- GARBAGE.
C      IF CONVERGED:
C        X -- THE CONVERGED VALUES OF THE CORRECTOR.
C        ERROR -- THE COMPONENT BY COMPONENT DIFFERENCE BETWEEN THE
C                 PREDICTED VALUES OF X AND THE CONVERGED VALUES.
C        COUNT -- THE NUMBER OF ITERATIONS REQUIRED FOR CONVERGENCE.
C      IF NOT CONVERGED:
C        X,ERROR,COUNT -- GARBAGE.
C
C
       INTEGER I,J
C
       COUNT = 0
   20  CALL FUNCT(X,ZERO,TIME)
       DO 30 I = FNUMP1,DERS
         ZERO(I) = X(I+SIZE) - H * ZERO(I)
   30  CONTINUE
       DO 40 I = DERSP1,SIZE
         ZERO(I) = X(I) - ZERO(I)
   40  CONTINUE
       CALL SOLVE(MAXVRS,leq,JAC,ZERO,PIVOT)
       DO 60 J = 1,leq
         DO 50 I = leqp1,SIZE
           ZERO(I) = ZERO(I) - JAC(I,J) * ZERO(J)
   50    CONTINUE
   60  CONTINUE
       CONV = .TRUE.
       DO 70 I = 1,SIZE
         X(I) = X(I) - L * ZERO(I)
         X(I+SIZE) = X(I+SIZE) - ZERO(I)
         ERROR(I) = ERROR(I) + ZERO(I)
         IF (CONV) THEN
           CONV = ABS(ZERO(I)) .LE. CTOL * XMAX(I)
         END IF
   70  CONTINUE
       COUNT = COUNT + 1
       IF (.NOT. CONV .AND. COUNT .LT. MAXCOR) GO TO 20
       RETURN
       END
       SUBROUTINE EVJAC(LEQ,X,EPS,XMAX,TIME,FNUM,FNUMP1,DERS,
      1 H,DERSP1,L,SIZE,MAXVRS,PIVOT,SING,JAC,Z,ZERO)
       IMPLICIT INTEGER (A-Z)
       INTEGER LEQ,SIZE,FNUMP1,DERS,DERSP1,MAXVRS,FNUM,PIVOT(*)
       REAL*8 X(*),EPS,XMAX(*),TIME,H,JAC(MAXVRS,*),L,Z(*),ZERO(*)
       LOGICAL SING
C
```

```
C THIS SUBROUTINE USES DIFFERENCES TO APPROXIMATE THE JACOBIAN MATRIX
C AND DECOMPOSES THE MATRIX INTO UPPER TRIANGULAR FORM.
C TO CHOOSE THE SIGN OF THE INCREMENT FOR EACH VARIABLE, THE SUBROUTINE
C EXAMINES THE VALUES OF THE FUNCTIONS AT THE CURRENT ITERATION VALUES
C AND TRIES TO DETERMINE IN WHICH DIRECTION THE CONVERGED VALUE OF EACH
C VARIABLE LIES.
C
C INPUT:
C      LEQ -- NUMBER OF JACOBIAN COLUMNS TO EVALUATE.
C      X -- THE CURRENT ITERATION VALUES FOR THE VARIABLES.
C      EPS -- THE ERROR TOLERANCE.
C      XMAX -- ARRAY CONTAINING A SIGNIFICANT VALUE FOR EACH OF THE
C              VARIABLES IN THE SYSTEM.
C      TIME -- THE INDEPENDENT VARIABLE OF THE SYSTEM.
C      FNUM -- POINTER TO THE LAST ROOT FINDING VARIABLE IN X.
C      FNUMP1 -- POINTER TO THE FIRST DIFFERENTIAL EQUATION VARIABLE IN
C                THE ARRAY X.
C      DERS -- POINTER TO THE LAST DIFFERENTIAL EQUATION VARIABLE IN
C              THE ARRAY X.
C      H -- SIZE OF THE CURRENT TIME STEP.
C      DERSP1 -- POINTER TO THE FIRST FIXED POINT VARIABLE IN THE ARRAY
C                X.
C      L -- A SCALING FACTOR FOR THE JACOBIAN WHICH DEPENDS ON THE
C           ORDER OF THE METHOD IN USE.
C      SIZE -- NUMBER OF EQUATIONS CURRENTLY IN THE SYSTEM.
C      MAXVRS -- DECLARED ROW DIMENSION OF JAC.
C      PIVOT -- THE PIVOT VECTOR USED BY DECOMP.
C      Z -- CONTENTS IGNORED. (USED AS WORK SPACE.  MUST BE DIMENSIONED
C           AT LEAST MAXVRS.)
C      ZERO -- CONTENTS IGNORED. (USED AS WORK SPACE. MUST BE
C              DIMENSIONED AT LEAST MAXVRS.)
C
C OUTPUT:
C      JAC -- THE JACOBIAN MATRIX SCALED BY A FACTOR OF L.
C      SING -- SET TO TRUE IF THE MATRIX IS SINGULAR TO THE WORKING
C              PRECISION OF THE MACHINE.
C      Z,ZERO -- GARBAGE.
C
$INCLUDE 9,IO6.CMN
      INTEGER I,J
      REAL*8 SUM,DELTA
      CHARACTER VARNAM*10
C
C GET VALUES OF FUNCTIONS WITH CURRENT ITERATION VALUES.
      CALL FUNCT(X,ZERO,TIME)
      DO 20 I=FNUMP1,DERS
        ZERO(I) = X(I+SIZE) - H*ZERO(I)
   20 CONTINUE
      DO 30 I=DERSP1,SIZE
        ZERO(I) = X(I) - ZERO(I)
   30 CONTINUE
C USE DIFFERENCES TO APPROXIMATE THE JACOBIAN.
      DO 70 I=1,leq
        SUM = X(I)
        DELTA = 0.D0
        IF (H.NE.0.D0) DELTA = HMIN*ABS(X(I+SIZE))/H
        IF (ABS(DELTA)*1.D8 .LT. XMAX(I)) DELTA = XMAX(I)/1.D8
        X(I) = SUM + DELTA
        CALL FUNCT(X,Z,TIME)
        DO 40 J = FNUMP1,DERS
          Z(J) = X(J+SIZE) - H * Z(J)
   40   CONTINUE
        DO 50 J = DERSP1,SIZE
```

```
          Z(J) = X(J) - Z(J)
   50   CONTINUE
        DO 60 J = 1,SIZE
          JAC(J,I) = L * (Z(J) - ZERO(J))/DELTA
   60   CONTINUE
        X(I) = SUM
   70 CONTINUE
      DO 80 I=FNUMP1,DERS
        JAC(I,I) = JAC(I,I) + 1.D0
   80 CONTINUE
C
C DECOMPOSE THE MATRIX INTO UPPER TRIANGULAR FORM.
      CALL DECOMP(MAXVRS,LEQ,JAC,DELTA,PIVOT,Z)
      WRITE(IWDSK,120) DELTA
      WRITE(IWTTY,120) DELTA
  120 FORMAT(X,'CONDITION NUMBER IS ',1PE10.3)
      SUM = DELTA + 1.D0
      SING = SUM.EQ.DELTA
      RETURN
      END
      SUBROUTINE UNPDSC(X,TIME,TDISC,FINISH,DSCTYP,CHNGOB,CHNGRM)
      IMPLICIT INTEGER(A-Z)
      REAL*8 TDISC,X(*),TIME,FINISH
      INTEGER DSCTYP,CHNGOB,CHNGRM
C
C      THIS SUBROUTINE IS CALLED AFTER THE COMPLETION OF
C A TIME STEP TO DETERMINE IF ANY UNPLANNED
C DISCONTINUITIES OCCURRED IN THAT TIME STEP.
C   A DISCONTINUITY IS A PARTICULAR
C TYPE OF INTERRUPTION  WHICH IS SPECIAL IN THAT IT
C REQUIRES A CHANGE IN THE SET OF EQUATIONS TO BE SOLVED.
C SO FAR, ALL UNPLANNED INTERRUPTIONS ARE
C DISCONTINUITIES.  EXAMPLES OF DISCONTINUITIES INCLUDE
C IGNITIONS AND BURNOUTS.  BOTH TYPES OF EVENT REQUIRE A
C CHANGE IN THE SET OF EQUATIONS TO BE SOLVED.
C     THE SUBROUTINE MUST PERFORM SEVERAL FUNCTIONS:
C  1. DETECTION -- THE SUBROUTINE EXAMINES THE
C      CONVERGED DATA FROM THE PREVIOUS TIME STEP TO
C      DETERMINE WHAT TYPES OF UNPLANNED DISCONTINUITIES
C      HAVE OCCURRED.  CURRENTLY, THE ONLY
C      DISCONTINUITIES WHICH ARE RECOGNIZED ARE IGNITION
C      (THROUGH HEATING OR CONTACT) BURN-OUT, AND
C      THE SPILLOVER OF HOT GASSES INTO A PREVIOUSLY UNINVOLVED
C      ROOM.
C  2. LOCATION -- AS THE SUBROUTINE DETECTS EACH
C      DISCONTINUITY IT MUST ESTIMATE AT WHAT TIME IT
C      OCCURRED.  CURRENTLY, THIS IS ACCOMPLISHED VIA
C      LINEAR EXTRAPOLATION.
C  3. SELECTION OF THE FIRST DISCONTINUITY -- IF MORE THAN
C      ONE DISCONTINUITY HAS OCCURRED DURING THE PREVIOUS
C      TIME STEP, THE SUBROUTINE MUST DECIDE WHICH ONE OCCURRED
C      FIRST.
C  4. SET FLAGS -- THE SUBROUTINE SETS TDISC TO THE TIME OF THE
C      FIRST DISCONTINUITY WHICH OCCURRED DURING THE JUST COMPLETED
C      TIME STEP, SETS DSCTYP TO INDICATE THE TYPE OF DISCONTIN-
C      UITY WHICH HAS OCCURRED AND SETS CHNGOB AND/OR CHNGRM
C      TO INDICATE WHICH ENTITY IS AFFECTED
C      BY THE STATE CHANGE.
C      NOTE THAT SETTING TDISC TO THE TIME OF
C      THE DISCONTINUITY TELLS THE MAIN PROGRAM TO REPEAT THE
C      TIME STEP WITH A PLANNED DISCONTINUITY AT TIME TDISC.
C
C             PLANNED VS. UNPLANNED INTERRUPTIONS
```

```
C       DURING ANY RUN, CERTAIN INTERRUPTIONS MAY OCCUR WHICH
C REQUIRE ADJUSTMENT OF THE TIME STEP.  FOR EXAMPLE, IF
C WE WANT OUTPUT AT A TIME T(C), THEN WE MUST ADJUST THE
C SIZE OF THE TIME STEP SO THAT THE NUMERICS CONVERGES
C AT TIME T(C).  SIMILARLY, IF AN OBJECT IGNITES AT TIME
C T(I), WE MUST MAKE SURE THAT THE PROGRAM CONVERGES AT
C TIME T(I) SINCE A NEW FIRE IN THE ROOM MAY CAUSE A
C DISCONTINUOUS CHANGE IN THE PHYSICS OF THE ROOM.
C OTHER POSSIBLE INTERRUPTIONS INCLUDE THE START-UP OF A
C DEBUGGER PACKAGE, THE OPENING OF A NEW VENT, AND THE
C ACTIVATION OF A SPRINKLER SYSTEM.
C      WE FIND IT NECESSARY TO DISTINGUISH BETWEEN TWO
C DIFFERENT TYPES OF INTERRUPTION, "PLANNED" AND
C "UNPLANNED".  SUPPOSE THAT THE PROGRAM HAS CONVERGED
C AT A TIME T(C) AND THAT AN INTERRUPTION IS TO OCCUR
C AT A TIME T(I)=T(C)+DT.  IF, AT TIME T(C) THE PROGRAM
C CAN FORSEE THAT AN INTERRUPTION WILL OCCUR AT TIME
C T(D), WE SAY THAT THE INTERRUPTION IS PLANNED.  IN
C THIS CASE, THE PROGRAM WILL BE CAREFUL NOT TO STEP
C BEYOND THE POINT OF THE INTERRUPTION.  AN EXAMPLE OF A
C PLANNED INTERRUPTION IS A BREAK FOR OUTPUT.  SINCE THE
C PROGRAM HAS A FIXED OUTPUT INTERVAL, TTYOUT, IT CAN
C PLAN A DISCONTINUITY WHENEVER T(D)=N*TTYOUT, FOR ANY
C INTEGER N.
C      ON THE OTHER HAND, IF AT TIME T(C) THE PROGRAM
C CANNOT FORSEE THAT THERE WILL BE A DISCONTINUITY AT
C TIME T(D), WE SAY THAT THE DISCONTINUITY IS UNPLANNED.
C FOR EXAMPLE, AT TIME T(C), THE PROGRAM CANNOT TELL IF
C AN OBJECT WILL REACH ITS IGNITION TEMPERATURE DURING
C THE UPCOMING TIME STEP OR DECIDE IF THE FIRE WILL
C BECOME HOT ENOUGH TO SUDDENLY OPEN A HOLE IN THE
C CEILING.  IN THIS CASE, THE PORGRAM MUST STEP BEYOND
C THE TIME T(D) OF THE INTERRUPTION BEFORE IT CAN DETECT
C THE INTERRUPTION.
C
C INPUTS:
C      X -- VECTOR OF CONVERGED VARIABLES.
C      TIME -- THE TIME AT WHICH THE VARIABLES ARE CONVERGED.
C      TDISC -- > FINISH IF NO DISCONTINUITY IS SCHEDULED.
C               < FINISH IF A DISCONTINUITY IS SCHEDULED BUT HAS
C               NOT YET BEEN REACHED.  IN THE LATTER CASE, TDISC IS
C               THE TIME OF THE PREVIOUSLY SCHEDULED DISCONTINUITY.
C      FINISH -- THE STOPPING TIME FOR THE RUN.
C
C OUTPUTS:
C      TDISC -- THE TIME OF THE FIRST UNPLANNED DISCONTINUITY
C               WHICH OCCURRED DURING THE JUST COMPLETED TIME
C               STEP.
C      DSCTYP -- THE TYPE OF DISCONTINUITY WHICH OCCURRED AT
C                TIME TDISC:
C                DSCTYP = 1 (BRNOUT) --> BURNOUT
C                DSCTYP = 2 (HEATIG) --> IGNITION BY HEATING
C                DSCTYP = 3 (CNTCIG) --> IGNITION BY CONTACT
C                DSCTYP = 4 (SPLOVR) --> HOT GASES BEGIN TO FLOW INTO
C                     A PREVIOUSLY UNINVOLVED ROOM.
C      CHNGOB -- THE OBJECT NUMBER OF THE OBJECT AFFECTED BY THE STATE
C           CHANGE.
C CHNGRM -- THE ROOM NUMBER OF THE ROOM CONTAINING THE OBJECT AFFECTED
C           BY THE STATE CHANGE.
C
      SAVE
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,VAR6.CMN
```

```
$INCLUDE 9,BLDNG.CMN
$INCLUDE 9,DSCNAM.CMN
C
C BRNOTM -- WHEN A BURNING OBJECT HAS MASS < OR = BRNOTM,
C           THE OBJECT IS CONSIDERED TO HAVE BURNT OUT.
   REAL*8 BRNOTM
   PARAMETER (BRNOTM = 1.D-6)
C
   INTEGER R,O,O2
   REAL*8 TDISC1,D,SUM,VSEP
C
C THE FOLLOWING STATEMENT FUNCTION LINEARLY INTERPOLATES T BACK
C TO THE TIME AT WHICH X HAD THE VALUE XI GIVEN THAT T HAD THE
C VALUE T1 WHEN X WAS X1 AND T WAS T2 WHEN X WAS X2.
   REAL*8 T1,T2,X1,X2,XI,INTERP
   INTERP(T1,T2,X1,X2,XI) = (XI-X1) * (T2-T1)/(X2-X1) + T1
C
C IF THERE'S ALREADY A DISCONTINUITY SCHEDULED, DON'T TRY TO
C SCHEDULE ANOTHER ONE.
   IF (TDISC.LE.FINISH) RETURN
C
C CHECK EACH ROOM FOR SPILLOVER FROM ANOTHER ROOM.
   DO 300 R=1,NROOMS
    IF (LAYR(R)) GOTO 300
    IF (TMUSM(R).NE.0.D0) THEN
C SPILLOVER DETECTED:
     TDISC1 = OLDTIM
     IF (TDISC1.LT.TDISC) THEN
      TDISC = TDISC1
      DSCTYP = SPLOVR
      CHNGRM = R
     END IF
    END IF
 300 CONTINUE
C
C
C CHECK EACH OBJECT FOR IGNITIONS AND BURNOUT.
   DO 30 R=1,NROOMS
    DO 20 O=1,OBJS(R)
C CHECK FOR BURNOUT
     IF (ZMOZZ(O,R).LE.BRNOTM .AND. STATE(O,R).NE.CNSUM) THEN
      TDISC1 = INTERP(OLDTIM,TIME,ZMOZZP(O,R),
  1          ZMOZZ(O,R),BRNOTM)
      IF (TDISC1.LT.TDISC) THEN
       TDISC = TDISC1
       DSCTYP = BRNOUT
       CHNGRM = R
       CHNGOB = O
      END IF
C CHECK FOR IGNITION BY HEATING
     ELSE IF (ZKOZZ(O,R).GE.ZKOIG(O,R).AND.STATE(O,R).NE.FLAME) THE
  1N
      TDISC1 = INTERP(OLDTIM,TIME,ZKOZZP(O,R),
  1               ZKOZZ(O,R),ZKOIG(O,R))
      IF (TDISC1.LT.TDISC) THEN
       TDISC = TDISC1
       DSCTYP = HEATIG
       CHNGRM = R
       CHNGOB = O
      END IF
C CHECK FOR IGNITION BY CONTACT
     ELSE
      DO 10 O2=1,OBJS(R)
```

```
              IF (O2.EQ.0) GO TO 10
              IF (STATE(O,R).EQ.FLAME .OR. STATE(O2,R).NE.FLAME) GO
     10
              D = SQRT((ZXOZZ(O,R)-ZXOZZ(O2,R))**2 +
     1             (ZYOZZ(O,R)-ZYOZZ(O2,R))**2)
C
C (3 APRIL 1981) IF WE ASSUME THAT OBJECTS ARE CIRCULAR, THEN THEY
C MAY SEEM TO TOUCH EVEN THOUGH THEY MAY NOT IN REALITY.  PROVISION-
C ALLY, THEREFORE, ASSUME OBJECTS ARE SQUARE.  IN FACT, HOWEVER,
C SHOULD HAVE IT IN THE INPUT WHETHER OBJECTS TOUCH OR OVERLAP!
C
              D = 1.12838D0*D
              SUM = ZRFZZ(O,R) + ZRFZZ(O2,R)
              IF (SUM.LT.D) GO TO 10
              VSEP = ZHOZZ(O2,R)-ZHOZZ(O,R)
              IF (VSEP .GT. ZNOZZ(O2,R)) GO TO 10
              IF (VSEP .LT. -ZHFZZ(O2,R)-ZNOZZ(O,R)) GO TO 10
C FOR CONTACT IGNITION, WE DON'T TRY TO INTERPOLATE FOR THE TIME
C OF IGNITION.
              TDISC1 = TIME
              IF (TDISC1.LT.TDISC) THEN
                TDISC = TDISC1
                CHNGRM = R
                CHNGOB = O
                DSCTYP = CNTCIG
              END IF
   10       CONTINUE
          END IF
   20   CONTINUE
   30 CONTINUE
      RETURN

      END


      SUBROUTINE CORCON(FNUM,SIZE,ZERO,CTOL,XMAX)
      IMPLICIT INTEGER (A-Z)
      INTEGER FNUM,SIZE
      REAL*8 ZERO(*),CTOL,XMAX(*)
C
C SUBROUTINE WRITES OUT A LIST OF ALL THE VARIABLES WHICH FAIL TO
C MEET THE CORRECTOR CONVERGENCE CRITERION.
C
C INPUTS:
C     FNUM -- NUMBER OF ROOT FINDING EQUATIONS IN THE SYSTEM.
C     SIZE -- NUMBER OF VARIABLES IN THE SYSTEM.
C     ZERO -- ARRAY CONTAINING THE INCREMENTS JUST APPLIED
C             TO THE VARIABLES TO OBTAIN THE NEW ITERATION
C             VALUES.
C     CTOL -- THE ERROR TOLERENCE FOR THE CORRECTOR.
C     XMAX -- ARRAY CONTAINING A SIGNIFICANT VALUE FOR EACH VARIABLE.
C
$INCLUDE 9,IO6.CMN
      CHARACTER*10 VARNAM
      INTEGER I,J
C
      WRITE(IWTTY,10) CTOL
      WRITE(IWDSK,10) CTOL
   10 FORMAT(/X,'SCALED CORRECTOR ERRORS (COMPARE AGAINST
     1 CTOL =',1PE8.1,')'/X,' VARIABLE ',X,'COR. ER.')
      DO 30 I=1,SIZE
        IF (ABS(ZERO(I)) .GT. CTOL*XMAX(I)) THEN
          WRITE(IWTTY,20) VARNAM(I),ZERO(I)/XMAX(I)
          WRITE(IWDSK,20) VARNAM(I),ZERO(I)/XMAX(I)
   20     FORMAT(X,A10,X,1PE8.1)
        END IF
   30 CONTINUE
      WRITE(IWTTY,40)
      WRITE(IWDSK,40)
   40 FORMAT(X)
      RETURN
      END
      SUBROUTINE DECOMP(MAXVRS,N,A,COND,PIVOT,WORK)
      IMPLICIT INTEGER (A-Z)
      INTEGER MAXVRS,N,PIVOT(N)
      REAL*8 A(MAXVRS,N),COND,WORK(N)
C
C  FROM FORSYTHE, MALCOLM, AND MOLER, "COMPUTER METHODS FOR MATHEMATICAL
C  COMPUTATIONS", P. 52, 1977.
C
C DECOMPOSES A REAL MATRIX BY GAUSSIAN ELIMINATION
C AND ESTIMATES THE CONDITION OF THE MATRIX.
C
C USE SOLVE TO COMPUTE SOLUTIONS TO LINEAR SYSTEMS.
C
C INPUT..
C   MAXVRS = DECLARED ROW DIMENSION OF THE ARRAY CONTAINING A.
C   N = ORDER OF THE MATRIX.
C   A = MATRIX TO BE TRAINGULARIZED.
C
C OUTPUT..
C   A CONTAINS AN UPPER TRIANGULAR MATRIX U AND A PERMUTED
C     VERSION OF A LOWER TRIANGULAR MATRIX I-L SO THAT
C     (PERMUTATION MATRIX)*A = L*U.
C   COND = AN ESTIMATE OF THE CONDITION OF A.
C     FOR THE LINEAR SYSTEM A*X = B, CHNAGES IN A AND B
C     MAY CAUSE CHANGES COND TIMES AS LARGE IN X.
C     IF COND+1.0.EQ.COND, A IS SINGULAR TO WORKING
C     PRECISION. COND IS SET TO 1.0E32 IF EXACT
C     SINGULARITY IS DETECTED.
C   PIVOT = THE PIVOT VECTOR.
C     PIVOT(K) = THE INDEX OF THE K-TH PIVOT ROW.
C     PIVOT(N) = (-1)**(NUMBER OF INTERCHANGES.
C
C WORK SPACE. THE VECTOR WORK MUST BE DECLARED AND INCLUDED
C     IN THE CALL. ITS INPUT CONTENTS ARE IGNORED.
C     ITS OUTPUT CONTENTS ARE USUALLY UNIMPORTANT.
C
C THE DETERMINANT OF A CAN BE OBTAINED ON OUTPUT BY
C   DET(A) = PIVOT(N)*A(1,1)*A(2,2)*...*A(N,N).
C
      INTEGER NM1,I,J,K,KP1,KB,KM1,M
      REAL*8 EK,T,ANORM,YNORM,ZNORM
C
      PIVOT(N)=1
      IF (N.EQ.1) GOTO 150
      NM1=N-1
C
C COMPUTE 1-NORM OF A
C
      ANORM=0.0D0
      DO 20 J=1,N
        T=0.0D0
        DO 10 I=1,N
          T=T+ABS(A(I,J))
   10   CONTINUE
```

```
          IF (T.GT.ANORM) ANORM=T
   20 CONTINUE
C
C GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
C
      DO 70 K=1,NM1
        KP1=K+1
C
C FIND PIVOT
C
        M=K
        DO 30 I=KP1,N
          IF (ABS(A(I,K)).GT.ABS(A(M,K))) M=I
   30   CONTINUE
        PIVOT(K)=M
        IF (M.NE.K) PIVOT(N)=-PIVOT(N)
        T=A(M,K)
        A(M,K)=A(K,K)
        A(K,K)=T
C
C SKIP STEP IF PIVOT IS ZERO
C
        IF (T.EQ.0.0D0) GOTO 70
C
C COMPUTE MULTIPLIERS
C
        DO 40 I=KP1,N
          A(I,K)=-A(I,K)/T
   40   CONTINUE
C
C INTERCHANGE AND ELIMINATE BY COLUMNS
C
        DO 60 J=KP1,N
          T=A(M,J)
          A(M,J)=A(K,J)
          A(K,J)=T
          IF (T.EQ.0.0D0) GOTO 60
          DO 50 I=KP1,N
            A(I,J)=A(I,J)+A(I,K)*T
   50     CONTINUE
   60   CONTINUE
   70 CONTINUE
C
C COND = (1-NORM OF A)*(AN ESTIMATE OF 1-NORM OF A-INVERSE)
C ESTIMATE OBTAINED BY ONE STEP OF INVERSE ITERATION FOR THE
C SMALL SINGULAR VECTOR. THIS INVOLVES SOLVING TWO SYSTEMS
C OF EQUATIONS, (A-TRANSPOSE)*Y = E AND A*Z = Y WHERE E
C IS A VECTOR OF +1 OR -1 CHOSEN TO CAUSE GROWTH IN Y.
C ESTIMATE = (1-NORM OF Z)/(1-NORM OF Y)
C
      DO 100 K=1,N
        T=0.0D0
        IF (K.EQ.1) GOTO 90
        KM1=K-1
        DO 80 I=1,KM1
          T=T+A(I,K)*WORK(I)
   80   CONTINUE
   90   EK=1.0D0
        IF (T.LT.0.0D0) EK=-1.0D0
        IF (A(K,K).EQ.0.0D0) GOTO 160
        WORK(K)=-(EK+T)/A(K,K)
  100 CONTINUE
      DO 120 KB=1,NM1
```

```
        K=N-KB
        T=0.0D0
        KP1=K+1
        DO 110 I=KP1,N
          T=T+A(I,K)*WORK(K)
  110   CONTINUE
        WORK(K)=T
        M=PIVOT(K)
        IF (M.EQ.K) GOTO 120
        T=WORK(M)
        WORK(M)=WORK(K)
        WORK(K)=T
  120 CONTINUE
C
      YNORM=0.0D0
      DO 130 I=1,N
        YNORM=YNORM+ABS(WORK(I))
  130 CONTINUE
C
C SOLVE A*Z=Y
C
      CALL SOLVE(MAXVRS,N,A,WORK,PIVOT)
C
      ZNORM=0.0D0
      DO 140 I=1,N
        ZNORM=ZNORM+ABS(WORK(I))
  140 CONTINUE
C
C ESTIMATE CONDITION
C
      COND=ANORM*ZNORM/YNORM
      IF (COND.LT.1.0D0) COND=1.0D0
      RETURN
C
C 1-BY-1
C
  150 COND=1.0D0
      IF (A(1,1).NE.0.0D0) RETURN
C
C EXACT SINGULARITY
C
  160 COND=1.0D32
      RETURN
      END
      SUBROUTINE LOOKUP(PTR,VARNUM,INDEX1,INDEX2)
      IMPLICIT INTEGER (A-Z)
      INTEGER PTR,VARNUM,INDEX1,INDEX2
C
C TAKES A POINTER TO A WORD IN THE FIRST SECTION OF COMMON
C /VAR/ AND RETURNS AN INTEGER CORRESPONDING TO THE ARRAY
C IN /VAR/ CONTAINING THE WORD AND INDICES INDICATING THE
C PARTICULAR WORD IN THE ARRAY.  THE SUBROUTINE ASSUMES SECTION
C 1 OF /VAR/ IS ORGANIZED IN A SPECIAL WAY AS FOLLOWS:
C
C     SECTION 1 OF /VAR/ CONTAINS 5 TYPES OF ARRAYS: ROOM
C     ARRAYS (ARRAYS INDEXED BY ROOM, XXXXX(MAXRMS)),
C     OBJECT ARRAYS (ARRAYS INDEXED BY OBJECT AND ROOM,
C     XXXXX(MAXOBS,MAXRMS)),  WALL ARRAYS (ARRAYS INDEXED BY WALL,
C     XXXXX(MAXWLS)), SIDE ARRAYS (ARRAYS INDEXED BY SIDE AND
C     WALL, XXXXX(2,MAXWLS)),AND VENT ARRAYS (ARRAYS INDEXED BY VENT,
C     XXXXX(MAXVTS)).  ALL ARRAYS OF A GIVEN TYPE ARE STORED
C     CONTIGUOUSLY.  THE FIVE TYPES OF ARRAY ARE STORED IN THE ORDER
C     GIVEN ABOVE.
```

```
C
C INPUT:
C     PTR -- A POINTER TO ONE OF THE WORDS IN THE FIRST
C            SECTION OF COMMON /VAR/.
C
C OUTPUT:
C     VARNUM -- THE NUMBER CORRESPONDING TO THE ARRAY CONTAIN-
C               ING THE WORD IN THE FIRST SECTION OF COMMON
C               /VAR/.  THE ARRAYS IN /VAR/ ARE NUMBERED IN THE
C               ORDER IN WHICH THEY ARE STORED.
C     INDEX1 -- THE FIRST INDEX OF THE PARTICULAR WORD IN
C               THE ARRAY.
C     INDEX2 -- THE SECOND INDEX OF THE PARTICULAR WORD IN
C               THE ARRAY.  IF THE ARRAY IS 1 DIMENSIONAL,
C               INDEX2 IS SET TO 0.  NOTE THAT THIS SUBROUTINE
C               ASSUMES THAT NO ARRAY IN /VAR/ HAS MORE THAN 2
C               DIMENSIONS.
C
      SAVE
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,IO6.CMN
      INTEGER RMSIZE,OBSIZE,WLSIZE,SDSIZE,VTSIZE,RMOFST,OBOFST,
     1 WLOFST,SDOFST,VTOFST
      PARAMETER (RMSIZE = MAXRMS,      OBSIZE = MAXRMS*MAXOBS,
     1           WLSIZE = MAXWLS,      SDSIZE = 2*MAXWLS,
     2           VTSIZE = MAXVTS,
     3           RMOFST = 0, OBOFST = RMOFST + RMVARS*RMSIZE,
     4           WLOFST = OBOFST + OBVARS*OBSIZE,
     5           SDOFST = WLOFST + WLVARS*WLSIZE,
     6           VTOFST = SDOFST + SDVARS*SDSIZE)
      INTEGER OFFSET,ARRSIZ
C
      IF (PTR .GT. NVARS .OR. PTR .LT. 1) THEN
        WRITE(IWTTY,10) PTR
   10   FORMAT(' ILLEGAL POINTER IN LOOKUP.  PTR=',I5)
      END IF
      OFFSET = PTR - 1
      IF (OFFSET .GE. VTOFST) THEN
        VARNUM = RMVARS + OBVARS + WLVARS + SDVARS + 1
        OFFSET = OFFSET - VTOFST
        ARRSIZ = VTSIZE
        INDEX1 = MAXVTS
        INDEX2 = 0
      ELSE IF (OFFSET .GE. SDOFST) THEN
        VARNUM = RMVARS + OBVARS + WLVARS + 1
        OFFSET = OFFSET - SDOFST
        ARRSIZ = SDSIZE
        INDEX1 = 2
        INDEX2 = MAXWLS
      ELSE IF (OFFSET .GE. WLOFST) THEN
        VARNUM = RMVARS + OBVARS + 1
        OFFSET = OFFSET - WLOFST
        ARRSIZ = WLSIZE
        INDEX1 = MAXWLS
        INDEX2 = 0
      ELSE IF (OFFSET .GE. OBOFST) THEN
        VARNUM = 1 + RMVARS
        OFFSET = OFFSET - OBOFST
        ARRSIZ = OBSIZE
        INDEX1 = MAXOBS
        INDEX2 = MAXRMS
      ELSE IF (OFFSET .GE. RMOFST) THEN
        VARNUM = 1
        OFFSET = OFFSET - RMOFST
        ARRSIZ = RMSIZE
        INDEX1 = MAXRMS
        INDEX2 = 0
      END IF
C
      VARNUM = VARNUM + OFFSET/ARRSIZ
      OFFSET = MOD(OFFSET,ARRSIZ)
      IF (INDEX2 .EQ. 0) THEN
        INDEX1 = OFFSET + 1
      ELSE
        INDEX2 = OFFSET / INDEX1 + 1
        INDEX1 = MOD(OFFSET,INDEX1) + 1
      END IF
      RETURN
      END
      SUBROUTINE NEWSTP(HMAX,TIME,X)
      IMPLICIT INTEGER (A-Z)
      REAL*8 HMAX,TIME,X(*)
C
C+    PART OF MODULE INTERFACE
C
C THIS SUBROUTINE SHOULD BE CALLED AT THE BEGINNING OF EACH TIME
C STEP TO PERFORM THE FOLLOWING INITIALIZATIONS:
C     1. CALL INITIALIZATION ENTRIES FOR THOSE PHYSICAL SUBROUTINES
C        WHICH PERFORM INTERNAL INTEGRATIONS (TMPW1 AND TMPO2).
C     2. SAVE THOSE VALUES FROM THE JUST COMPLETED TIME STEP WHICH
C        THE PHYSICS WILL NEED IN THE CALCULATIONS FOR THE NEXT
C        TIME STEP.
C
$INCLUDE 9/DIMN.CMN
$INCLUDE 9/IO6.CMN
$INCLUDE 9/VAR6.CMN
$INCLUDE 9/BLDNG.CMN
      REAL*8 H,ZKA(2)
      INTEGER W,R,O,I
      LOGICAL FIRST
      DATA FIRST/.TRUE./
C
C H IS THE SIZE OF THE TIME STEP WHICH HAS JUST BEEN COMPLETED.
      CALL UNPACK(X)
      IF (.NOT. FIRST) THEN
        H = TIME - OLDTIM
      ELSE
        FIRST = .FALSE.
      END IF
      OLDTIM = TIME
      DO 20 W=1,NWALLS
        DO 10 I=1,2
          IF (WLINK(I,W).EQ.0) THEN
            ZKA(I) = ZKAOUT
          ELSE
            ZKA(I) = ZKAZZ(WLINK(I,W))
          END IF
   10   CONTINUE
C
C NOTE THAT AT THE FIRST TIME STEP, SEVERAL OF THE ARGUMENTS IN THE
C CALLS TO TMPW1I AND TMPO2I ARE UNDEFINED.  HOWEVER, NEITHER ENTRY
C USES ANY OF THE UNDEFINED ARGUMENTS AT THE FIRST TIME STEP.
        CALL TMPW1I(W,ZGWZZ(W),HMAX,ZNWZZ(W),IWTTY,ZKA(1),ZKA(2),
     1   ZJWZZ(W),ZKWZZ(1,W),ZKWZZ(2,W),H)
   20 CONTINUE
C
```

```
      DO 40 R=1,NROOMS
        DO 30 O=1,OBJS(R)
          IF (STATE(O,R) .NE. HEATNG) GO TO 30
          CALL TMPO2I(MAXOBS*(R-1)+O,ZGOZZ(O,R),HMAX,ZNOZZ(O,R),
     1     ZJOZZ(O,R),ZKAZZ(R),H,ZKOZZ(O,R))
   30   CONTINUE
   40 CONTINUE
C
      DO 60 R=1,NROOMS
        DO 50 O=1,OBJS(R)
          ZRFZZP(O,R) = ZRFZZ(O,R)
          ZHFZZP(O,R) = ZHFZZ(O,R)
          TMOZZP(O,R) = TMOZZ(O,R)
          ZMOZZP(O,R) = TMOZZ(O,R)
          ZKOZZP(O,R) = ZKOZZ(O,R)
   50   CONTINUE
   60 CONTINUE
C
      RETURN
      END
      SUBROUTINE PACKX(X)
      IMPLICIT INTEGER (A-Z)
      REAL*8 X(*)
C
C THIS SUBROUTINE PACKS THE SPARSELY DISTRIBUTED VARIABLES FROM
C SECTION 1 OF /VAR/ INTO THE COMPACT VECTOR X WHICH IS SENT TO
C THE NUMERICS.  THE POINTER ARRAY INSYS IS USED TO SELECT WHICH
C VARIABLES IN /VAR/ ARE PACKED INTO X.
C
      SAVE
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,VAR6.CMN
      INTEGER INSYS
      COMMON /SYSPTR/ INSYS(MAXVRS+1)
      REAL*8 XIN(NVARS)
      EQUIVALENCE (XIN,TELZR)
      INTEGER INDEX
      INDEX = 1
C WHILE INSYS(INDEX)<>0 DO
   10 IF (INSYS(INDEX).EQ.0) GO TO 20
        X(INDEX) = XIN(INSYS(INDEX))
        INDEX = INDEX + 1
        GO TO 10
   20 CONTINUE
      RETURN
      END
      SUBROUTINE SOLVE(MAXVRS,N,A,B,PIVOT)
      IMPLICIT INTEGER (A-Z)
      INTEGER MAXVRS,N,PIVOT(N)
      REAL*8 A(MAXVRS,N),B(N)
C
C  FROM FORSYTHE, MALCOLM, AND MOLER, "COMPUTER METHODS FOR MATHEMATICAL
C  COMPUTATIONS", P. 55, 1977.
C
C SOLUTION OF LINEAR SYSTEM, A*X = B
C DO NOT USE IF DECOMP HAS DETECTED SINGULARITY.
C
C INPUT..
C   MAXVRS = DECLARED ROW DIMENSION OF ARRAY CONTAINING A.
C   N = ORDER OF THE MATRIX.
C   A = TRIANGUALIZED MATRIX OBTAINED FROM DECOMP.
C   B = RIGHT HAND SIDE VECTOR.
C   PIVOT = PIVOT VECTOR OBTAINED FORM DECOMP.
```

```
C
C OUTPUT..
C   B = SOLUTION VECTOR, X.
C
      INTEGER KB,KM1,NM1,KP1,I,K,M
      REAL*8 T
C
C FORWARD ELIMINATION
C
      IF (N.EQ.1) GOTO 50
      NM1 = N-1
      DO 20 K=1,NM1
        KP1=K+1
        M=PIVOT(K)
        T=B(M)
        B(M)=B(K)
        B(K)=T
        DO 10 I=KP1,N
          B(I)=B(I)+A(I,K)*T
   10   CONTINUE
   20 CONTINUE
C
C BACK SUBSTITUTION
C
      DO 40 KB=1,NM1
        KM1=N-KB
        K=KM1+1
        B(K)=B(K)/A(K,K)
        T=-B(K)
        DO 30 I=1,KM1
          B(I)=B(I)+A(I,K)*T
   30   CONTINUE
   40 CONTINUE
   50 B(1)=B(1)/A(1,1)
      RETURN
      END
      SUBROUTINE UNPACK(X)
      IMPLICIT INTEGER (A-Z)
C
C+      PART OF MODULE INTERFACE
C
      REAL*8 X(*)
C
C THIS SUBROUTINE UNPACKS THE TIGHTLY PACKED VECTOR X INTO THE
C SPARSE ARRAYS IN THE FIRST SECTION OF /VAR/ USING THE POINTERS
C IN THE ARRAY INSYS TO PLACE THE SUCCESSIVE COMPONENTS OF X
C INTO THE CORRECT PLACES IN /VAR/.
C
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,VAR6.CMN
      INTEGER INDEX
      REAL*8 XIN(NVARS)
      INTEGER INSYS
      COMMON /SYSPTR/ INSYS(MAXVRS+1)
C
C NOTE THAT THE FOLLOWING EQUIVALENCE ASSUMES THAT TELZR IS THE FIRST
C ARRAY IN COMMON /VAR/.
      EQUIVALENCE (XIN,TELZR)
C
      INDEX = 1
C
C WHILE INSYS(INDEX)<>0 DO
   10 IF (INSYS(INDEX).EQ. 0 ) GO TO 20
```

```
        XIN(INSYS(INDEX)) = X(INDEX)
        INDEX = INDEX + 1
        GO TO 10
   20 CONTINUE
      RETURN
C
      END
      CHARACTER*(*) FUNCTION VARNAM(PTR)
      IMPLICIT INTEGER (A-Z)
      INTEGER PTR
C
C RETURNS THE ALPHANUMERIC NAME OF THE PTR'TH COMPONENT
C IN THE VECTOR OF VARIABLES CURRENTLY IN THE SYSTEM.
C
C INPUT:
C     PTR -- THE INDEX OF A VARIABLE IN THE VECTOR OF
C            OF VARIABLES CURRENTLY IN THE SYSTEM.  THE
C            ORDER OF THE COMPONENTS IN THIS VECTOR IS
C            DEFINED BY THE ORDER OF THE POINTERS IN THE
C            ARRAY INSYS.
C
C OUTPUT:
C     VARNAM -- STRING OF 10 CHARACTERS GIVING THE ARRAY NAME
C               AND SUBSCRIPTS OF THE GIVEN VARIABLE.
C
$INCLUDE 9/DIMN.CMN
      INTEGER INSYS
      COMMON /SYSPTR/ INSYS(MAXVRS+1)
      INTEGER VARNUM,INDEX1,INDEX2
      CHARACTER AA
      CHARACTER*5 NAMES(RMVARS+OBVARS+WLVARS+SDVARS+VTVARS)
      DATA NAMES/'TELZR','TELZO','ZMLZZ','TMLZZ','ZELZZ',
     1 'TELZZ','ZHLZZ','ZKLZZ','ZYLOZ','ZYLDZ','ZYLMZ',
     2 'ZYLSZ','ZYLWZ','ZPRZZ','ZULZZ','TEPSM','TEOSM',
     3 'TMODS','TMOMS','TMOSS','TMOWS',' ZMDL',
     4 ' ZMML',' ZMOL',' ZMWL',' ZMSL','AVLSM',' WVLS',
     5 ' PPLR','TMOSM','TMPSM','TMUSM','TMDSM','TEUSM',
     6 'EPLSM','BOXYS','TMDZF','TMWZF','TMSZF','TMMZF',
     + 'TMOZF','FCPOR','ZKOZZ','ZMOZZ',
     7 'TMOZZ','TEOZZ','ZHPZZ','TMPZZ',
     8 'TEPZR','ZRFZZ','ZHFZZ','FCLOR',
     9 'FCWOR','TMOZB','  PHI','ACWZZ','ZKWZZ',
     1 'FCLWR','FCPWR','FCLWD'/
C
      CALL LOOKUP(INSYS(PTR),VARNUM,INDEX1,INDEX2)
C THE FOLLOWING LINE GETS THE NAME OF THE VARIABLE AND FILLS OUT VARNAM
C WITH SPACES ON THE RIGHT.
      VARNAM(1:) = NAMES(VARNUM)(1:5)
      IF (INDEX1 .NE. 0) THEN
        VARNAM(6:6) = '('
C       WRITE(UNIT=VARNAM(7:7),FMT='(I1)') INDEX1
        ENCODE(AA,17) INDEX1
        VARNAM(7:7) = AA
   17   FORMAT(I1)
        IF (INDEX2 .EQ. 0) THEN
          VARNAM(8:8) = ')'
        ELSE
          VARNAM(8:8) = ','
C         WRITE(UNIT=VARNAM(9:9),FMT='(I1)') INDEX2
          ENCODE(AA,17) INDEX2
          VARNAM(9:9) = AA
          VARNAM(10:10) = ')'
        END IF
      END IF
      RETURN
      END
      SUBROUTINE WRIT(X,TIME,NT,IDEVIC)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER IDEVIC
      REAL*8 X(*)
C
C CALLED BY MAIN, DEBUG, NWTN, JACB.
C
C IN FILE W05281.FOR (FORMERLY WRITL2)
C CODED BY N.BILLIKOPF, L. TREFETHEN, AUGUST 1977
C MODIFIED BY M.RABUZZI,3-10-81
C
C WRITES THE INPUT VECTOR X ONTO DEVICE IDEVIC (TELETYPE OR DISK).
C THE POINTER INFORMATION IN COMMON BLOCK /POINTR/ DETERMINES WHICH
C ELEMENTS OF THE ARRAYS IN X ARE TO BE OUTPUT.
C
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,IO6.CMN
$INCLUDE 9,VAR6.CMN
$INCLUDE 9,BLDNG.CMN
C
C
C
C       IF (ISPOUT .NE. 0  .AND.  IDEVIC .EQ. IWDSK) THEN
C        CALL WRIT03
C        RETURN
C       END IF
C
C       IF (ICMODE .NE. 0 .AND. IDEVIC .EQ. IWTTY) THEN
C        ITABLE = 0
C        CALL GRAPH2(ITABLE)
C        IF (ITABLE .EQ. 0) RETURN
C       END IF
C
      CALL UNPACK(X)
      WRITE (IDEVIC,100)
      WRITE (IDEVIC,110)
      WRITE (IDEVIC,120) TIME,TIME-OLDTIM,NT,0,0,'ALDF'
C
C OUTPUT BY ROOM NUMBER:
      DO 10 KR=1,NROOMS
       WRITE (IDEVIC,40) KR,TELZR(KR),TELZD(KR),ZMLZZ(KR),TMLZZ(KR),
     1          ZELZZ(KR),TELZZ(KR),ZHLZZ(KR),ZKLZZ(KR),ZYLOZ(KR),
     2          ZYLDZ(KR),ZYLMZ(KR),ZYLSZ(KR),ZYLWZ(KR),ZPRZZ(KR)
C
C OUTPUT BY OBJECT NUMBER WITHIN THAT ROOM:
C
        DO 10 KO=1,OBJS(KR)
C
C OUTPUT OBJECT INFO IF OBJECT NOT BURNT-OUT (STATE < 10),
C  PLUME INFO IF OBJECT PYROLYZING (STATE >= 3),
C   FIRE RADIUS IF OBJECT BURNING (STATE >= 4)
C
          IF (STATE(KO,KR) .LT. 10) THEN
           WRITE (IDEVIC,60) KO,FQLOR(KO,KR),FQWOR(KO,KR),FQPOR(KO,KR),
     1                   KO,ZKOZZ(KO,KR),
     2                   ZMOZZ(KO,KR),TMOZZ(KO,KR),TEOZZ(KO,KR)
            IF (STATE(KO,KR) .GE. 3) THEN
              CALL TEMPLM(ZKAZZ(KR),ZKOPY(KO,KR),TEOZZ(KO,KR),TEPZR(KO,K
     1R),                 TMOZZ(KO,KR),TMPZZ(KO,KR),TEPZZ)
              WRITE (IDEVIC,70) ZHPZZ(KO,KR),TMPZZ(KO,KR),TEPZZ,
     1                    TEPZR(KO,KR),ZHFZZ(KO,KR),TMOZB(KO,KR)
              IF (STATE(KO,KR) .GE. 4) THEN
                WRITE (IDEVIC,80) ZRFZZ(KO,KR)
              END IF
            END IF
          END IF
C
   10 CONTINUE
C
C OUTPUT BY VENT NUMBER:
C
      DO 20 KV=1,NVENTS
        KRI=VLINK(1,KV)
        KRO =VLINK(2,KV)
        IF (VLINK(1,KV).EQ.0) THEN
          CALL FLOW(CD,CP,(VMAOUT+VMAZZ(KRO))/2.D0,ZBVZZ(KV),ZHRZZ(KRO),
     1  ZHTZZ(KV),ZHVZZ(KV),ZKAOUT,ZKAZZ(KRO),0.D0,ZHLZZ(KRO),ZKAOUT,
     2  ZKLZZ(KRO),0.D0,ZPRZZ(KRO),TEUZZ,TMDZZ,TMUZZ)
        ELSE IF (KRO.EQ.0) THEN
          CALL FLOW(CD,CP,(VMAZZ(KRI)+VMAOUT)/2.D0,ZBVZZ(KV),ZHRZZ(KRI),
     1ZHTZZ(KV),ZHVZZ(KV),ZKAZZ(KRI),ZKAOUT,ZHLZZ(KRI),0.D0,ZKLZZ(KRI),
     2  ZKAOUT,ZPRZZ(KRI),0.D0,TEUZZ,TMDZZ,TMUZZ)
        ELSE
          CALL FLOW(CD,CP,(VMAZZ(KRI)+VMAZZ(KRO))/2.D0,ZBVZZ(KV),
     1 ZHRZZ(KRI),ZHTZZ(KV),ZHVZZ(KV),ZKAZZ(KRI),ZKAZZ(KRO),ZHLZZ(KRI),
     2 ZHLZZ(KRO),ZKLZZ(KRI),ZKLZZ(KRO),ZPRZZ(KRI),ZPRZZ(KRO),
     3 TEUZZ,TMDZZ,TMUZZ)
        END IF
   20 WRITE (IDEVIC,50) KV,TEUZZ,TMUZZ,TMDZZ
C
C OUTPUT BY WALL NUMBER AND SIDE:
      DO 30 KW=1,NWALLS
   30 WRITE (IDEVIC,90) (KW,JSIDE,FQLWR(JSIDE,KW),FQPWR(JSIDE,KW),
     1          FQLWD(JSIDE,KW),ZKWZZ(JSIDE,KW),JSIDE=1,2)
C
      WRITE (IDEVIC,110)
      RETURN
C
   40 FORMAT(' ROOM=',I2,':',T14,'TELZR=',1PE11.4,3X,'TELZD=',1PE11.4,
     1          3X,'ZMLZZ=',1PE11.4/T14,'TMLZZ=',1PE11.4,
     2          3X,'ZELZZ=',1PE11.4,3X,'TELZZ=',1PE11.4,/,
     3          T14,'ZHLZZ='1PE11.4,3X,'ZKLZZ=',1PE11.4,
     4          3X,'ZYLOZ=',1PE11.4,/,T14,'ZYLDZ=',1PE11.4,
     5          3X,'ZYLMZ=',1PE11.4,3X,'ZYLSZ=',1PE11.4,/,
     6          T14,'ZYLWZ=',1PE11.4,3X,'ZPRZZ=',1PE11.4)
   50 FORMAT(' VENT=',I2,':',T14,'TEUZZ=',1PE11.4,3X,'TMUZZ=',1PE11.4,
     1          3X,'TMDZZ=',1PE11.4)
   60 FORMAT(3X,'OBJ=',I2,':',T14,'FQLOR=',1PE11.4,3X,'FQWOR=',
     1          1PE11.4,3X,'FQPOR=',1PE11.4,/,3X,'(ID=',I2,')',T14,
     2          'ZKOZZ=',1PE11.4,3X,'ZMOZZ=',1PE11.4,3X,
     3          'TMOZZ=',1PE11.4,/,T14,'TEOZZ=',1PE11.4)
   70 FORMAT(T14,'ZHPZZ=',1PE11.4,3X,'TMPZZ=',1PE11.4,
     1          3X,'TEPZZ=',1PE11.4,/,T14,'TEPZR=',1PE11.4,3X,
     1          'ZHFZZ=',1PE11.4,3X,'TMOZB=',1PE11.4)
   80 FORMAT(T14,'ZRFZZ=',1PE11.4)
   90 FORMAT(' WALL=',I2,',',I1,':',T14,'FQLWR=',1PE11.4,
     1          3X,'FQPWR=',1PE11.4,
     2          3X,'FQLWD=',1PE11.4,/,T14,'ZKWZZ=',1PE11.4)
  100 FORMAT(' ')
  110 FORMAT(1X,70('-'))
  120 FORMAT(1X,'T=',F8.3,5X,'DT=',F6.3,5X,'NT=',I4,5X,
     1          'NIT=',I5,5X,'IT=',I3,5X,A4)
      END
      BLOCK DATA SCALER
      IMPLICIT INTEGER (A-Z)
C
C THIS DATA BLOCK INITIALIZES THE VALUES IN COMMON /SCALER/:
C
$INCLUDE 9,DIMN.CMN
C
      REAL*8 TELZRA,TELZDA,ZMLZZA,TMLZZA,ZELZZA,
     1 TELZZA,ZHLZZA,ZKLZZA,ZYLOZA,ZYLDZA,ZYLMZA,
     2 ZYLSZA,ZYLWZA,ZPRZZA,ZULZZA,TEPSMA,TECSMA,
     3 TMODSA,TMOMSA,TMOSSA,TMOWSA,ZMDLA,
     4 ZMMLA,ZMOLA,ZMWLA,ZMSLA,AVLSMA,WVLSA,
     5 PPLRA,TMOSMA,TMPSMA,TMUSMA,TMDSMA,TEUSMA,
     6 EPLSMA,BOXYSA,TMDZFA,TMWZFA,TMSZFA,TMMZFA,
     + TMOZFA,FQPORA,ZKOZZA,ZMOZZA,
     7 TMOZZA,TEOZZA,ZHPZZA,TMPZZA,
     8 TEPZRA,ZRFZZA,ZHFZZA,FCLORA,
     9 FCWORA,TMOZBA,PHIA,AQWZZA,ZKWZZA,
     1 FCLWRA,FQPWRA,FCLWDA
C
C COMMON /SCALER/ CONTAINS CONSTANTS THAT REPRESENT THE SMALLEST
C VALUE OF SIGNIFICANCE OF VARIABLES IN COMMON /VAR/.
      COMMON /SCALER/ TELZRA(MR), TELZDA(MR), ZMLZZA(MR),
     1 TMLZZA(MR), ZELZZA(MR), TELZZA(MR), ZHLZZA(MR), ZKLZZA(MR),
     2 ZYLOZA(MR), ZYLDZA(MR), ZYLMZA(MR), ZYLSZA(MR), ZYLWZA(MR),
     3 ZPRZZA(MR), ZULZZA(MR), TEPSMA(MR), TECSMA(MR),
     4 TMODSA(MR), TMOMSA(MR),
     5 TMOSSA(MR), TMOWSA(MR), ZMDLA(MR), ZMMLA(MR), ZMOLA(MR),
     6 ZMWLA(MR), ZMSLA(MR), AVLSMA(MR), WVLSA(MR), PPLRA(MR),
     7 TMOSMA(MR), TMPSMA(MR), TMUSMA(MR), TMDSMA(MR), TEUSMA(MR),
     8 EPLSMA(MR), BOXYSA(MR), TMDZFA(MR), TMWZFA(MR), TMSZFA(MR),
     + TMMZFA(MR), TMOZFA(MR), FQPORA(MO,MR),  ZKOZZA(MO,MR),
     9 ZMOZZA(MO,MR), TMOZZA(MO,MR), TEOZZA(MO,MR), ZHPZZA(MO,MR),
     1 TMPZZA(MO,MR), TEPZRA(MO,MR), ZRFZZA(MO,MR), ZHFZZA(MO,MR),
     2 FCLORA(MO,MR), FQWORA(MO,MR), TMOZBA(MO,MR), PHIA(MO,MR),
     3 AQWZZA(MW), ZKWZZA(2,MW), FQLWRA(2,MW), FQPWRA(2,MW),
     4 FCLWDA(2,MW)
C
C COMMON /SCALER/ INITIALIZATION.
      INTEGER TOTOB,WSIDES,TOBM2
      PARAMETER (TOTOB = MAXOBS*MAXRMS, WSIDES = 2*MAXWLS)
      PARAMETER (TOBM2 = TOTOB - 2)
      DATA TELZRA/MR*1.D3/,  TELZDA/MR*1.D3/,  ZMLZZA/MR*1.0D-1/
      DATA TMLZZA/MR*1.0D-3/, ZELZZA/MR*1.0D4/,  TELZZA/MR*1.0D3/
      DATA ZHLZZA/MR*1.0D-3/, ZKLZZA/MR*0.1D0/,    ZYLOZA/MR*1.0D-3/
      DATA ZYLDZA/MR*1.0D-4/, ZYLMZA/MR*1.0D-6/, ZYLSZA/MR*1.0D-4/
      DATA ZYLWZA/MR*1.0D-4/, ZPRZZA/MR*1.0D-5/,ZULZZA/MR*1.0D-3/
      DATA TEPSMA/MR*1.0D2/,    TMODSA/MR*1.D-5/,  TMOMSA/MR*1.D-7/
      DATA TECSMA/MR*1.0D2/
      DATA TMOSSA/MR*1.D-5/,   TMOWSA/MR*1.D-5/,   ZMDLA/MR*1.0D-2/
      DATA ZMMLA/MR*1.0D-6/,   ZMOLA/MR*1.0D-2/,   ZMWLA/MR*1.0D-3/
      DATA ZMSLA/MR*1.0D-5/,   AVLSMA/MR*1.0D0/,  WVLSA/MR*1.0D-1/
      DATA PPLRA/MR*1.D1/,       TMOSMA/MR*1.D-5/,   TMPSMA/MR*1.D-3/
      DATA TMUSMA/MR*1.D-3/,   TMDSMA/MR*1.D-3/,   TEUSMA/MR*100.D0/
      DATA EPLSMA/MR*1.D-3/,   BOXYSA/MR*1.D-6/,
     1   FQPORA/TOTOB*1.D3/
      DATA ZKOZZA/TOTOB*.1D0/, ZMOZZA/TOTOB*1.D-6/,TMOZZA/TOTOB*1.D-5/
      DATA TEOZZA/TOTOB*1.D2/, ZHPZZA/TOTOB*1.D-3/,TMPZZA/TOTOB*1.D-3/
      DATA TEPZRA/TOTOB*1.D2/, ZRFZZA/TOTOB*1.D-3/,ZHFZZA/TOTOB*1.D-2/
      DATA FCLORA/TOTOB*1.D3/, FCWORA/TOTOB*1.D3/, TMOZBA/TOTOB*1.D-6/
      DATA PHIA/TOTOB*5.D2/,      AQWZZA/MAXWLS*5.D0/
      DATA ZKWZZA/WSIDES*.1D0/,FCLWRA/WSIDES*1.D3/,FQPWRA/WSIDES*1.D3/
      DATA FCLWDA/WSIDES*1.D3/
```

```
C
      END




      SUBROUTINE GRAPH2(ITABLE)
C
C CALLED BY WRIT
C
C CODED BY JOHN ROCKETT, APRIL 1981.
C ADAPTED FOR MARK VI BY ANDREW KAHNG, AUGUST 1982.
C
C PROVIDES A GRAPHICAL DISPLAY OF THE ROOM WITH SELECTED DATA
C SUPERIMPOSED.  MORE SOPHISTICATED THAN GRAPH1.  AS YET DOES
C NOT HAVE THE ABILITY TO GENERATE HARD COPY.
C
C INCLUDE 'DIMN.INC'
C INCLUDE 'IO.INC'
C INCLUDE 'VAR.CMN'
C INCLUDE 'BLDNG.CMN'
C
C
C CHARACTER*1 BLANK,DOT,COLON,OH,STAR,PRINT(71),LW
C DATA BLANK,DOT,COLON,OH,STAR /' ','.',':','O','*'/
C
C PI = 4.*ATAN(1.)
C TO = 273.16
C
C THE NEXT TWO LINES SET THE SMOKE LEVELS AT WHICH VARIOUS
C SYMBOLS WILL BE USED.  THESE LEVELS ARE ARBITRARY AND MAY
C BE ARBITRARILY CHANGED WITHOUT AFFECTING PROGRAM PERFORMANCE.
C
C ZYLS1 = 0.001
C ZYLS2 = 0.002
C FLH1 = 57324.1
C
C RROOM = SQRT((ZLRZX(1)*ZLRZY(1))/PI)
C DELL = RROOM/58.
C DELH = ZHRZZ(1)/15.
C MFUEL = IFIX(REAL(ZRFZI(1,1)/DELL + 0.5))
C MFIRE = 0
C IF(ABS(TMOZZ(1,1)).GT.0.) MFIRE = IFIX(REAL(ZRFZZ(1,1)/DELL + 0.5))
C IF (MFIRE .GT. MFUEL) MFIRE = MFUEL
C FLAMEH = (ABS(TEOZZ(1,1)/FLH1))**0.4
C NFLAME = IFIX(REAL((ZHRZZ(1)-ZHCZZ(1,1)-FLAMEH)/DELH + 0.5))
C
C IF(NLAYR .GE. NFLAME) NFLAME = NLAYR + 1
C
C IF(NFLAME.EQ.NFUEL) NFLAME = NFUEL-1
C NTRANS = IFIX(REAL(ZHTZZ(1)/DELH + 0.5))
C NSILL = IFIX(REAL((ZHTZZ(1)+ZHVZZ(1))/DELH + 0.5))
C NFUEL = IFIX(REAL((ZHRZZ(1)-ZHCZZ(1,1))/DELH + 0.5))
C IF(NFUEL.GT.14) NFUEL = 14
C NMASS = NFUEL+1
C NFUEL1 = IFIX(REAL((ZHRZZ(1)-ZHOZZ(1,1)+ZNOZZ(1,1))/DELH + 0.5))
C IF(NFUEL1.EQ.NFUEL)NFUEL1 = NFUEL+1
C NLAYR = IFIX(REAL(ZHLZZ(1)/DELH + 0.5))
C NLAYRT = 0
C IF (NLAYR.GT.1) NLAYRT = NLAYR/2
C
C
```

```
C DUM = ZKCZZ(1) - TO
C WRITE(IWTTY,10) ZLRZX(1),ZLRZY(1),ZLRZZ(1),DUM
C 10 FORMAT(5X,' LENGTH =',F6.2,' M',3X,' WIDTH =',F6.2,' M',
C    1  3X,' HEIGHT =',F6.2,' M',3X,' TAMB =',F6.2,' C')
C WRITE(IWTTY,30)
C DUM = ABS(TEOZZ(1,1)+TEOZZ(2,1)) * 0.001
C WRITE(6,20) ZTZZZ,DUM
C 20 FORMAT(5X,' TIME =',F5.0,' SEC.',3X,' HEAT RELEASE RATE =',
C    1 1PE9.2,' KW')
C WRITE(6,30)
C 30 FORMAT(' ')
C
C N = 1
C DUM = ZKWZZ(1,1) - TO
C WRITE(6,40) DUM
C 40 FORMAT(5X,'************************************************',
C    1 '********** TWALL =',F4.0,' C ****')
C
C SUM = ZKLZZ(1) - TO
C DUM1 = 100.*ZYLOZ(1)
C1050 CONTINUE
C
C IF LINE TO BE PRINTED IS ABOVE SOFFIT, INCLUDES LEFT WALL
C IF LINE TO BE PRINTED IS IN HOT LAYER, INCLUDES SMOKE
C IF HOT LAYER ABOVE SOFFIT, NO SMOKE TO LEFT OF WALL
C
C IF(N.LE.NLAYR) THEN
C  IF(ZYLSZ(1).LT.ZYLS1) THEN
C   DO 1100 I = 1,71
C1100   PRINT(I) = DOT
C  ELSE
C   IF((ZYLSZ(1).LT.ZYLS2).AND.(ZYLSZ(1).GE.ZYLS1)) THEN
C   DO 1101 I = 1,71
C1101   PRINT(I) = COLON
C  ELSE
C   DO 1103 I = 1,71
C1103   PRINT(I) = OH
C END IF
C END IF
C IF(N.LE.NTRANS) PRINT(14) = STAR
C IF(NLAYR.LE.NTRANS) THEN
C DO 1110 I = 1,13
C1110 PRINT(I) = BLANK
C END IF
C IF(N.GT.NLAYRT) THEN
C DO 1111 I = 1,5
C1111 PRINT(I) = BLANK
C END IF
C IF(N.EQ.NLAYR) PRINT(6) = BLANK
C IF(N.GE.(NSILL-1)) THEN
C DO 1120 I = 1,19
C1120 PRINT(I) = BLANK
C END IF
C IF(N.GE.NSILL) THEN
C PRINT(14) = STAR
C PRINT(20) = BLANK
C END IF
C
C CAPTION SMOKE LAYER
C
C IF(N.EQ.NLAYRT) THEN
C WRITE(6,531) (PRINT(I),I=14,18),DUM,
C    1 (PRINT(I),I=35,37),DUM1,(PRINT(I),I=55,71)
```

```
C GOTO 1055
C END IF
C IF(N.EQ.(NLAYRT+1)) THEN
C WRITE(IWTTY,532)TMUZZ(1),(PRINT(I),I=14,71)
C GOTO 1055
C END IF
C IF((N.NE.NLAYRT).OR.(N.NE.(NLAYRT+1))) THEN
C WRITE(IWTTY,521) PRINT
C GOTO 1055
C END IF
C
C ELSE
C
C LINE TO BE PRINTED IS BELOW THE LAYER, PRINT PLUME (OR FLAME)
C WIDTH OF PLUME AND HEIGHT OF FLAME REFLECT FIRE SIZE
C
C LW = BLANK
C IF(N.LE.NTRANS) LW = STAR
C IF(N.GE.NSILL) LW = STAR
C CALL GPLUME(N,NFLAME,NFUEL,MFIRE,MFUEL,ZMOZZ(1,1),LW)
C GOTO 1055
C END IF
C1055 CONTINUE
C N = N+1
C IF(N.LT.15) GOTO 1050
C 521 FORMAT(5X,71A1,'*')
C 531 FORMAT(5X,' VENT FLOW   ',5A1,' TLAYER =',F4.0,' C ',
C     1 3A1,' OXYGEN =',F5.1,' % ',17A1,'*')
C 532 FORMAT(1PE12.1,' KG/S ',58A1,'*')
C WRITE(6,190)
C 190 FORMAT(5X,'*************!***************************************',
C     1 '*********************************')
C DUM = ((FQLOR(2,1)+FQWOR(2,1)+FQPOR(2,1)))*0.001
C DUM1 = ABS(TMOZZ(1,1))
C WRITE(6,200)DUM,DUM1
C 200 FORMAT(10X,' OBJ. 2 FLUX =',1PE10.2,' KW/M**2',4X,'BURN RATE =',
C     1 1PE10.2,' KG/SEC')
C WRITE(6,30)
C1220 CONTINUE
C        WRITE(6,220)
C 220 FORMAT(' TABULAR OUTPUT?  (ANS. N OR Y)')
C READ(5,230)ICHAR
C 230 FORMAT(A1)
C IF(ICHAR.EQ.'Y') ITABLE = 1
C IF(ICHAR.EQ.'Y') RETURN
C IF((ICHAR.EQ.'N').OR.(ICHAR.EQ.' ')) RETURN
C GOTO 1220
        RETURN
C
C           2/10/83---THIS MODULE WAS DISABLED BY JOHN LUTZ AND JAY MIRASOL
C                     AT THE REQUEST OF HENRI MITLER.
C
      END

C
C
      SUBROUTINE GPLUME(N,NFLAME,NFUEL,MFIRE,MFUEL,ZMOZ,LW)
C
C CALLED BY GRAPH2
C
C CODED BY JOHN ROCKETT, APRIL 1981.
C ADAPTED FOR MARK VI BY ANDREW KAHNG, AUGUST 1982.
C
```

```
C CALLED BY GRAPHW TO CONSTRUCT A REPRESENTATION OF THE BURNING
C OBJECT AND ITS PLUME ON THE TERMINAL SCREEN.
C
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,IO6.CMN
$INCLUDE 9,VAR6.CMN
$INCLUDE 9,BLONG.CMN
C
      CHARACTER*1 LW,PRINT(58)
      CHARACTER*1 STAR,O1,BLANK,EXCL,PLUME,FIRE
      DATA STAR,O1,BLANK,EXCL,PLUME,FIRE
     1 /'*','1',' ','!','P','F'/

      MFI = MFIRE
      IF(MFIRE.GT.48) MFI = 48
      MFU = MFUEL
      IF(MFUEL.GT.48) MFU = 48
      DO 1010 I = 1,57
 1010 PRINT(I) = BLANK
      PRINT(58) = STAR
      FM = FLOAT(MFI)
      FO = FLOAT(MFU)
      ML = IFIX((FM+1)*0.5)
C         MR = IFIX(FM*0.5)     <--   THIS DOESN'T COMPILE
      MOL = IFIX(FO*0.5)
      MOR = IFIX((FO-1)*0.5)
C
      IF(N.LT.NFLAME) THEN
      IF(ML.EQ.0) GOTO 1100
      IF(ML.NE.0) THEN
      LL = 36-ML
      LR = 35-ML+MFI
      DO 1030 I = LL,RR
 1030 PRINT(I) = PLUME
      END IF
      GOTO 1100
C
      ELSE
      IF((N.GE.NFLAME).AND.(N.LT.NFUEL)) THEN
      IF(ML.EQ.0) GOTO 1100
      IF(ML.NE.0) THEN
      LL = 36-ML
      LR = 35-ML+MFI
      DO 1040 I = LL,RR
 1040 PRINT(I) = FIRE
      END IF
      GOTO 1100
      END IF
C
      IF(N.EQ.NFUEL) THEN
      LI = 35-MOL
      LR = 35+MOR
      PRINT(LI) = EXCL
      DO 1050 I = LI+1,LR-1
 1050 PRINT(I) = O1
      PRINT(LR) = EXCL
      GOTO 1100
      END IF
      END IF
C
      IF(N.EQ.(NFUEL+1)) WRITE(IWTTY,100) LW,ZMOZ
      IF(N.GT.(NFUEL+1)) WRITE(IWTTY,110) LW
      RETURN
```

```
C
 1100 WRITE(6,70) LW,PRINT
      RETURN
   70 FORMAT(18X,A1,58A1)
  100 FORMAT(18X,A1,26X,' FUEL=',F6.2,' KG',16X,'*')
  110 FORMAT(18X,A1,57X,'*')
      END
```

```
      SUBROUTINE ERFDE(TIME)
      IMPLICIT INTEGER (A-Z)
      REAL*8 TIME
C
C THIS SUBROUTINE IS CALLED BY FUNCT TO EVALUATE THE FUNCTIONS
C ASSOCIATED WITH THE ROOT FINDING AND DIFFERENTIAL EQUATIONS.
C ERFDE (EVALUATE ROOT FINDING AND DIFFERENTIAL EQUATIONS) CALLS
C THE PHYSICAL SUBROUTINES PASSING INPUT VARIABLES FROM THE FIRST
C SECTION OF COMMON /VAR/.  THE OUTPUT OF EACH PHYSICAL SUBROUTINE
C ASSOCIATED WITH A ROOT FINDING EQUATION IS PLACED INTO THE SECOND
C SECTION OF /VAR/. THE OUTPUT OF EACH PHYSICAL SUBROUTINE
C ASSOCIATED WITH A DIFFERENTIAL EQUATION IS PLACED IN THE THIRD
C SECTION OF /VAR/.  SINCE THE ARRAYS USED TO STORE INPUT VARIABLES
C ARE ISOLATED FROM THE ARRAYS USED TO STORE OUTPUT VARIABLES,
C THE ORDER IN WHICH THE FUNCTIONS ARE EVALUATED IS NOT SIGNIFICANT.
C
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,BLDNG.CMN
$INCLUDE 9,IC6.CMN
$INCLUDE 9,VAR6.CMN
      INTEGER FIXED,REGIM1,REGIM2,REGIM3
      PARAMETER (FIXED=0,REGIM1=1,REGIM2=2,REGIM3=3)
C
      INTEGER FTELZR,FTELZD,FZMLZZ,FTMLZZ,FZELZZ,
     1 FTELZZ,FZHLZZ,FZKLZZ,FZYLOZ,FZYLDZ,FZYLMZ,
     2 FZYLSZ,FZYLWZ,FZPRZZ,FZULZZ,FTEPSM,FTEOSM,
     3 FTMODS,FTMOMS,FTMOSS,FTMOWS,FZMDL,
     4 FZMML,FZMOL,FZMWL,FZMSL,FAVLSM,FWVLS,
     5 FPPLR,FTMOSM,FTMPSM,FTMUSM,FTMDSM,FTEUSM,
     6 FEPLSM,FBOXYS,FTMDZF,FTMWZF,FTMSZF,FTMMZF,
     + FTMOZF,FFQPOR,FZKOZZ,FZMOZZ,
     7 FTMOZZ,FTEOZZ,FZHPZZ,FTMPZZ,
     8 FTEPZR,FZRFZZ,FZHFZZ,FFQLOR,
     9 FFQWOR,FTMOZB,FPHI,FAQWZZ,FZKWZZ,
     + FFQLWR,FFQPWR,FFQLWD
C
      COMMON /VARYNG/ FTELZR(MR),FTELZD(MR),FZMLZZ(MR),FTMLZZ(MR),
     1 FZELZZ(MR),FTELZZ(MR),FZHLZZ(MR),FZKLZZ(MR),
     2 FZYLOZ(MR),FZYLDZ(MR),FZYLMZ(MR),FZYLSZ(MR),
     3 FZYLWZ(MR),FZPRZZ(MR),FZULZZ(MR),FTEPSM(MR),
     4 FTEOSM(MR),FTMODS(MR),FTMOMS(MR),FTMOSS(MR),
     5 FTMOWS(MR),FZMDL(MR),FZMML(MR),FZMOL(MR),
     6 FZMWL(MR),FZMSL(MR),FAVLSM(MR),FWVLS(MR),
     7 FPPLR(MR),FTMOSM(MR),FTMPSM(MR),FTMUSM(MR),
     8 FTMDSM(MR),FTEUSM(MR),FEPLSM(MR),FBOXYS(MR),
     9 FTMDZF(MR),FTMWZF(MR),FTMSZF(MR),FTMMZF(MR),
     + FTMOZF(MR),FFQPOR(MO,MR),FZKOZZ(MO,MR),
     1 FZMOZZ(MO,MR),FTMOZZ(MO,MR),FTEOZZ(MO,MR),
     2 FZHPZZ(MO,MR),FTMPZZ(MO,MR),FTEPZR(MO,MR),
     3 FZRFZZ(MO,MR),FZHFZZ(MO,MR),FFQLOR(MO,MR),
     4 FFQWOR(MO,MR),FTMOZB(MO,MR),FPHI(MO,MR),
     5 FAQWZZ(MW),FZKWZZ(2,MW),FFQLWR(2,MW),FFQPWR(2,MW),
     6 FFQLWD(2,MW)
C
      INTEGER R,O,W
C
C
C TMNET:
      DO 10 R=1,NROOMS
        IF (FZPRZZ(R).EQ.REGIM1) THEN
        CALL NETMAS(CP,ZKAZZ(R),TELZZ(R),TMDSM(R),TMLZZ(R),TMOSM(R),
     1      TMUSM(R),TMNET(R))
        END IF
   10 CONTINUE
C
      DO 30 R=1,NROOMS
        IF (FZMLZZ(R).EQ.REGIM1) DZMLZZ(R) = TMLZZ(R)
        IF (FZELZZ(R).EQ.REGIM1) DZELZZ(R) = TELZZ(R)
        IF (FZMDL(R).EQ.REGIM1)
     1      CALL TCO2M(TMODS(R),TMOSM(R),TMPSM(R),TMDZF(R),
     1             DZMDL(R))
        IF (FZMML(R).EQ.REGIM1)
     1        CALL TMNXM(TMOMS(R),TMMZF(R),DZMML(R))
        IF (FZMOL(R).EQ.REGIM1)
     1         CALL TOXYM(BOXYS(R),TMOSM(R),TMPSM(R),TMOZF(R),
     1             DZMOL(R))
        IF (FZMWL(R).EQ.REGIM1)
     1       CALL TH2OM(TMOWS(R),TMWZF(R),DZMWL(R))
        IF (FZMSL(R).EQ.REGIM1)
     1        CALL TSUTM(TMOSS(R),TMSZF(R),DZMSL(R))
        DO 20 O=1,OBJS(R)
          IF (FZMOZZ(O,R).EQ.REGIM1) DZMOZZ(O,R) = TMOZZ(O,R)
          IF (FZRFZZ(O,R).EQ.REGIM1) THEN
             CALL SPREAD(AFIRE(O,R),ZKFZZ,ZRFZM(O,R),ZRFZO(O,R),
     1       PHI(O,R),ZRFZZ(O,R),DZRFZZ(O,R))
          ELSE IF (FZRFZZ(O,R).EQ.REGIM2) THEN
            CALL PULRAD(ZRFZM(O,R),ZRFZO(O,R),ZTIG(O,R),TIME,
     1         ZRFZZ(O,R))
          END IF
   20   CONTINUE
   30 CONTINUE
C
      DO 40 W=1,NWALLS
        IF (FAQWZZ(W).EQ.REGIM1)
     1   DAQWZZ(W) = FQLWD(1,W) + FQLWD(2,W)
   40 CONTINUE
C
      RETURN
      END
      SUBROUTINE EVALFP(TIME)
      IMPLICIT INTEGER (A-Z)
      REAL*8 TIME
C
C+    PART OF MODULE INTERFACE
C
C THIS SUBROUTINE CALLS THE PHYSICAL SUBROUTINES TO EVALUATE THE
C FUNCTION ASSOCIATED WITH EACH FIXED POINT EQUATION.
C THE INPUT VARIABLES FOR THE PHYSICAL SUBROUTINES ARE TAKEN
C FROM THE FIRST SECTION OF /VAR/.  THE OUTPUT OF EACH PHYSICAL
C SUBROUTINE IS PLACED IN THE APPROPRIATE ARRAY IN THE FIRST
C SECTION OF /VAR/.  SINCE THE OUTPUT OF THE PHYSICAL SUBROUTINES
C IS PLACED INTO THE SAME ARRAYS FROM WHICH THE INPUT TO THE
C PHYSICAL SUBROUTINES IS TAKEN, THE ORDER OF FUNCTION EVALUATION
C IS IMPORTANT AND PROPER ORDERING CAN SIMPLIFY THE PROBLEM
C BY VARIABLE ELIMINATION.  THE ORDER OF EVALUATION IS DETERMINED
```

```
C BY THE ARRAY EVORD.
C
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,VAR6.CMN
$INCLUDE 9,BLDNG.CMN
$INCLUDE 9,IC6.CMN
      INTEGER FIXED,REGIM1,REGIM2,REGIM3
      PARAMETER (FIXED=0,REGIM1=1,REGIM2=2,REGIM3=3)
C
      INTEGER FTELZR,FTELZD,FZMLZZ,FTMLZZ,FZELZZ,
     1 FTELZZ,FZHLZZ,FZKLZZ,FZYLOZ,FZYLDZ,FZYLMZ,
     2 FZYLSZ,FZYLWZ,FZPRZZ,FZULZZ,FTEPSM,FTEOSM,
     3 FTMODS,FTMOMS,FTMOSS,FTMOWS,FZMDL,
     4 FZMML,FZMOL,FZMWL,FZMSL,FAVLSM,FWVLS,
     5 FPPLR,FTMOSM,FTMPSM,FTMUSM,FTMDSM,FTEUSM,
     6 FEPLSM,FBOXYS,FTMDZF,FTMWZF,FTMSZF,FTMMZF,
     + FTMOZF,FFQPOR,FZKOZZ,FZMOZZ,
     7 FTMOZZ,FTEOZZ,FZHPZZ,FTMPZZ,
     8 FTEPZR,FZRFZZ,FZHFZZ,FFQLOR,
     9 FFQWOR,FTMOZB,FPHI,FAQWZZ,FZKWZZ,
     + FFQLWR,FFQPWR,FFQLWD
C
      COMMON /VARYNG/ FTELZR(MR),FTELZD(MR),FZMLZZ(MR),FTMLZZ(MR),
     1 FZELZZ(MR),FTELZZ(MR),FZHLZZ(MR),FZKLZZ(MR),
     2 FZYLOZ(MR),FZYLDZ(MR),FZYLMZ(MR),FZYLSZ(MR),
     3 FZYLWZ(MR),FZPRZZ(MR),FZULZZ(MR),FTEPSM(MR),
     4 FTEOSM(MR),FTMODS(MR),FTMOMS(MR),FTMOSS(MR),
     5 FTMOWS(MR),FZMDL(MR),FZMML(MR),FZMOL(MR),
     6 FZMWL(MR),FZMSL(MR),FAVLSM(MR),FWVLS(MR),
     7 FPPLR(MR),FTMOSM(MR),FTMPSM(MR),FTMUSM(MR),
     8 FTMDSM(MR),FTEUSM(MR),FEPLSM(MR),FBOXYS(MR),
     9 FTMDZF(MR),FTMWZF(MR),FTMSZF(MR),FTMMZF(MR),
     + FTMOZF(MR),FFQPOR(MO,MR),FZKOZZ(MO,MR),
     1 FZMOZZ(MO,MR),FTMOZZ(MO,MR),FTEOZZ(MO,MR),
     2 FZHPZZ(MO,MR),FTMPZZ(MO,MR),FTEPZR(MO,MR),
     3 FZRFZZ(MO,MR),FZHFZZ(MO,MR),FFQLOR(MO,MR),
     4 FFQWOR(MO,MR),FTMOZB(MO,MR),FPHI(MO,MR),
     5 FAQWZZ(MW),FZKWZZ(2,MW),FFQLWR(2,MW),FFQPWR(2,MW),
     6 FFQLWD(2,MW)
C
      INTEGER EVSETS
C EVSETS IS THE NUMBER OF EVALUATION SETS IN THIS SUBROUTINE.
      PARAMETER (EVSETS = 42)
C EVORD IS THE ARRAY WHICH GIVES THE ORDER IN WHICH THE EQUATIONS
C ARE TO BE EVALUATED TO ACHIEVE MAXIMUM VARIABLE ELIMINATION.
      INTEGER EVORD(EVSETS+1)
      REAL*8 H,SUM,SUM1(8,MAXRMS),FQ,TMOD1,FQ1PW,TMOM1,TMOS1,
     1     TMOW1,AVL1,WVL1,P1PLR1,SIGN,TEU,TMD,TMU,TMPLU1,BOXY1,
     2     TEPZZ,ZHL,ZKA,ZUL,CO2FLO,H2OFLO,SUTFLO,MNXFLO,CXYFLO
      INTEGER I,R,O,O2,W,V,R1,R2,SIDE
C SPACE SAVING EQUIVALENCE
      EQUIVALENCE (FQ,TMOD1,FQ1PW,TMOM1,TMOS1,TMOW1,AVL1,WVL1,
     1             P1PLR1,TEU,TMPLU1,BOXY1,TEPZZ)
      DATA EVORD/17,22,5,6,8,9,7,10,12,13,19,15,14,32,18,20,
     1 31,33,25,34,2,21,24,35,1,41,11,38,4,37,3,27,36,29,
     2 28,40,30,23,26,42,16,39,0/
C
C H -- SIZE OF THE CURRENT TIME STEP
      H = TIME - OLDTIM
C
      I = 0
C
C MAIN LOOP BEGINS HERE:
   10 I = I + 1
      IF (EVCRD(I) .EQ. 0) RETURN
      GO TO (20,40,60,80,100,120,140,160,180,200,220,240,
     1       290,330,360,400,430,470,500,530,560,630,660,690,730,
     2       760,790,820,850,880,910,940,970,1000,1030,1060,1090,1120,
     3       1170,1200,1230,1260) EVCRD(I)
C
C
C FIXED POINT EQUATIONS:
C
C TELZR:
   20 DO 30 R=1,NROOMS
        IF (FTELZR(R).EQ.REGIM1) THEN
C
C NOTE THAT IN THE FOLLOWING CALL WE ASSUME THAT EACH ROOM HAS
C 1 WALL WITH SIDE 1 INSIDE AND SIDE 2 OUTSIDE.
        W = R
        CALL RDNL(ZKAZZ(R),ZLRZX(R),ZLRZY(R),AVLSM(R),FQPWR(1,W),
     1           PPLR(R),ZHLZZ(R),ZKLZZ(R),ZKWZZ(1,W),ZULZZ(R),
     2           TELZR(R))
        END IF
   30 CONTINUE
      GO TO 10
C
C TELZD:
   40 DO 50 R=1,NROOMS
        IF (FTELZD(R).EQ.REGIM1) THEN
C
C NOTE THAT IN THE FOLLOWING CALL WE ASSUME THAT EACH ROOM HAS
C 1 WALL WITH SIDE 1 INSIDE AND SIDE 2 OUTSIDE.
        W = R
        CALL CNVL1(ZLRZX(R),ZLRZY(R),AQWZZ(W),AVLSM(R),FQLWD(1,W),
     1            WVLS(R),TELZZ(R),ZELZZ(R),ZHLZZ(R),TELZD(R))
      END IF
   50 CONTINUE
      GO TO 10
C
C TMLZZ:
   60 DO 70 R=1,NROOMS
        IF (FTMLZZ(R).EQ.REGIM1) THEN
        CALL TMSL(VMAZZ(R),ZHRZZ(R),ZLRZX(R),ZLRZY(R),TMPSM(R),
     1           TMUSM(R),ZMLZZ(R),TMLZZ(R))
      END IF
   70 CONTINUE
      GO TO 10
C
C TELZZ:
   80 DO 90 R=1,NROOMS
        IF (FTELZZ(R).EQ.REGIM1) THEN
        CALL POWL(ZHRZZ(R),ZLRZX(R),ZLRZY(R),TELZD(R),TELZR(R),
     1     TEPSM(R),TEUSM(R),ZELZZ(R),TELZZ(R))
        END IF
   90 CONTINUE
      GO TO 10
C
C ZHLZZ:
  100 DO 110 R=1,NROOMS
        IF (FTELZZ(R).EQ.REGIM1) THEN
        CALL LAYDPT(CP,VMAZZ(R),ZHRZZ(R),ZKAZZ(R),ZLRZX(R),
     1     ZLRZY(R),ZELZZ(R),ZHLZZ(R))
        END IF
  110 CONTINUE
      GO TO 10
```

```
C
C ZKLZZ:
  120 DO 130 R=1,NROOMS
          IF (FZKLZZ(R).EQ.REGIM1) THEN
          CALL TMPL(ZKAZZ(R),CP,ZELZZ(R),
     1             ZMLZZ(R),ZKLZZ(R))
          END IF
  130 CONTINUE
      GO TO 10
C
C ZYLOZ:
  140 DO 150 R=1,NROOMS
          IF (FZYLCZ(R).EQ.REGIM1) THEN
          CALL OXYCNC(ZMOL(R),ZMLZZ(R),ZYLOZ(R))
          END IF
  150 CONTINUE
      GO TO 10
C
C ZYLDZ:
  160 DO 170 R=1,NROOMS
          IF (FZYLCZ(R).EQ.REGIM1) THEN
          CALL CXDCNC(ZMDL(R),ZMLZZ(R),ZYLCZ(R))
          END IF
  170 CONTINUE
      GO TO 10
C
C ZYLMZ:
  180 DO 190 R=1,NROOMS
          IF (FZYLMZ(R).EQ.REGIM1) THEN
          CALL MNXCNC(ZMML(R),ZMLZZ(R),ZYLMZ(R))
          END IF
  190 CONTINUE
      GO TO 10
C
C ZYLSZ:
  200 DO 210 R=1,NROOMS
          IF (FZYLSZ(R).EQ.REGIM1) THEN
          CALL SUTCNC(ZMSL(R),ZMLZZ(R),ZYLSZ(R))
          END IF
  210 CONTINUE
      GO TO 10
C
C ZYLWZ:
  220 DO 230 R=1,NROOMS
          IF (FZYLWZ(R).EQ.REGIM1) THEN
          CALL WTRCNC(ZMWL(R),ZMLZZ(R),ZYLWZ(R))
          END IF
  230 CONTINUE
      GO TO 10
C
C ZULZZ:
  240 DO 280 R=1,NROOMS
          IF (FZULZZ(R).EQ.REGIM1) THEN
          GO TO (250,260,270), VERSUN(2)
C USE ABSRB1, 2, 3 AS INDICATED BY VERSUN(2).
  250     CALL ABSRB1(TIME,ZULZZ(R))
          GO TO 280
  260     CALL ABSRB2(ZYLSZ(R),ZULZZ(R))
          GO TO 280
  270     CALL ABSRB3(ZKLZZ(R),ZYLCZ(R),ZYLMZ(R),ZYLOZ(R),ZYLSZ(R),
     1                ZULZZ(R))
          GO TO 280
          END IF
  280 CONTINUE
      GO TO 10
C
C FQLOR:
  290 DO 320 R=1,NROOMS
        DO 310 O=1,OBJS(R)
          IF (FFQLOR(O,R).EQ.REGIM1) THEN
          IF (ANGH(O,R) .GT. 45.D0) THEN
            WRITE (IWTTY,300)
  300       FORMAT(' RNLV HAS NOT YET BEEN INCORPORATED INTO THE CODE.')
            STOP
          ELSE
            CALL RNLH(STATE(O,R),ZHOZZ(O,R),ZHRZZ(R),ZLRZX(R),ZLRZY(R),
     1             ZUFZZ,ZXOZZ(O,R),ZYOZZ(O,R),ZHFZZ(O,R),ZHLZZ(R),
     2             ZKLZZ(R),ZRFZZ(O,R),ZULZZ(R),FQLOR(O,R))
          END IF
          END IF
  310   CONTINUE
  320 CONTINUE
      GO TO 10
C
C FQWOR:
  330 DO 350 R=1,NROOMS
        DO 340 O=1,OBJS(R)
          IF (FFQWOR(O,R).EQ.REGIM1) THEN
          CALL RNWO02(STATE(O,R),ZHOZZ(O,R),ZHRZZ(R),ZKAZZ(R),
     2               ZLRZX(R),ZLRZY(R),ZUFZZ,ZXOZZ(O,R),ZYOZZ(O,R),
     3               ZHFZZ(O,R),ZHLZZ(R),ZKWZZ(1,R),ZRFZZ(O,R),
     4               ZULZZ(R),FQWOR(O,R))
          END IF
  340   CONTINUE
  350 CONTINUE
      GO TO 10
C
C FQPOR:
  360 DO 390 R=1,NROOMS
        DO 380 O=1,OBJS(R)
          IF (FFQPOR(O,R).EQ.REGIM1) THEN
          SUM = 0.D0
          DO 370 O2=1,OBJS(R)
            FQ = 0.D0
            IF (STATE(O2,R).EQ.FLAME)
     1       CALL RNPO01(STATE(O,R),(R-1)*MAXOBS+O2,ZKFZZ,
     2         TMOZZP(O2,R),ZHFZZP(O2,R),ZHOZZ(O2,R),ZHOZZ(O,R),
     3         ZHRZZ(R),ZRFZI(O2,R),ZRFZI(O,R),ZUFZZ,ZXOZZ(O2,R),
     4         ZXOZZ(O,R),ZYOZZ(O2,R),ZYOZZ(O,R),TMOZZ(O2,R),
     5         ZHFZZ(O2,R),ZHLZZ(R),ZHPZZ(O2,R),ZMOZZ(O2,R),
     6         ZRFZZ(O2,R),ZRFZZ(O,R),FQ)
            SUM = SUM + FQ
            FQ = 0.D0
            IF (STATE(O,R).EQ.FLAME .AND. STATE(O2,R).EQ.FLAME
     1         .AND. O.NE.O2)
     2       CALL RNFFO2(ZKFZZ,ZHOZZ(O2,R),ZHOZZ(O,R),ZHRZZ(R),
     3         ZRFZI(O,R),ZRFZI(O2,R),ZUFZZ,ZXOZZ(O2,R),ZXOZZ(O,R),
     4         ZYOZZ(O2,R),ZYOZZ(O,R),ZHFZZ(O2,R),ZHFZZ(O,R),ZHLZZ(R),
     5         ZRFZZ(O2,R),ZRFZZ(O,R),FQ)
            SUM = SUM + FQ
  370     CONTINUE
          FQPOR(O,R) = SUM
          END IF
  380   CONTINUE
  390 CONTINUE
      GO TO 10
```

```
C
C ZKOZZ:
  400 DO 420 R=1,NROOMS
        DO 410 O=1,OBJS(R)
          IF (FZKOZZ(O,R).EQ.REGIM1) THEN
          CALL TMPO2(H,EB(O,R),(R-1)*MAXOBS+O,ZHOZZ(O,R),ZHRZZ(R),
     2      ZKAZZ(R),ZOAZZ,FQLOR(O,R),FQPOR(O,R),
     3      FQWOR(O,R),ZHLZZ(R),ZKLZZ(R),ZKOZZ(O,R),
     4      ZKOZZ(O,R))
          END IF
  410   CONTINUE
  420 CONTINUE
      GO TO 10
C
C TMOZZ:
  430 DO 460 R=1,NROOMS
        DO 450 O=1,OBJS(R)
          IF (FTMOZZ(O,R).EQ.REGIM1) THEN
              CALL PYRO(QVAP(O,R),ZRFZM(O,R),PHI(O,R),ZMOZZ(O,R),
     1              ZRFZZ(O,R),TMOZZ(O,R))
          ELSE IF (FTMOZZ(O,R).EQ.REGIM2) THEN
              WRITE (IWTTY,440)
  440         FORMAT(' BFIR HAS NOT YET BEEN INCORPORATED INTO THE CODE.
     1')
              STOP
          END IF
  450   CONTINUE
  460 CONTINUE
      GO TO 10
C
C TEOZZ:
  470 DO 490 R=1,NROOMS
        DO 480 O=1,OBJS(R)
          IF (FTEOZZ(O,R).EQ.REGIM1) THEN
           CALL CHEM(CP,QF(O,R),QVAP(O,R),ZKAZZ(R),ZKOPY(O,R),
     2            ZRFZM(O,R),TMOZB(O,R),ZHPZZ(O,R),TEOZZ(O,R))
          END IF
  480   CONTINUE
  490 CONTINUE
      GO TO 10
C
C ZHPZZ:
  500 DO 520 R=1,NROOMS
        DO 510 O=1,OBJS(R)
          IF (FZHPZZ(O,R).EQ.REGIM1) THEN
          CALL PLMHT(ZHOZZ(O,R),ZHRZZ(R),ZHLZZ(R),ZHPZZ(O,R))
          END IF
  510   CONTINUE
  520 CONTINUE
      GO TO 10
C
C TMPZZ:
  530 DO 550 R=1,NROOMS
        DO 540 O=1,OBJS(R)
          IF (FTMPZZ(O,R).EQ.REGIM1) THEN
          CALL TMSPLM(ALPHA,VMAZZ(R),ZKAZZ(R),TEOZZ(O,R),
     2       TMOZZ(O,R),ZHPZZ(O,R),ZRFZZ(O,R),TMPZZ(O,R))
          END IF
  540   CONTINUE
  550 CONTINUE
      GO TO 10
C
C TEPZR:
  560 DO 580 R=1,NROOMS
        DO 570 O=1,OBJS(R)
          IF (FTEPZR(O,R).EQ.REGIM1) THEN
          CALL RDNP(ZUFZZ,ZKFZZ,TIME,(R-1)*MAXOBS+O,ZRFZZ(O,R),
     2      ZHFZZP(O,R),ZLRZX(R),ZLRZY(R),ZHFZZ(O,R),
     3      ZMOZZ(O,R),TMOZZP(O,R),TMOZZ(O,R),TECZZ(O,R),
     4      TEPZR(O,R))
          END IF
  570   CONTINUE
  580 CONTINUE
      GO TO 10
C
C ZHFZZ:
  630 DO 650 R=1,NROOMS
        DO 640 O=1,OBJS(R)
          IF (FZHFZZ(O,R).EQ.REGIM1) THEN
          CALL HEIGHT(CHI(O,R),TPSIO(O,R),TMOZB(O,R),TMOZZ(O,R),
     2                ZRFZZ(O,R),ZHFZZ(O,R))
          END IF
  640   CONTINUE
  650 CONTINUE
      GO TO 10
C
C FQLWR:
  660 DO 680 W=1,NWALLS
        DO 670 SIDE = 1,2
          IF (FFQLWR(SIDE,W).EQ.REGIM1) THEN
          R = WLINK(SIDE,W)
          CALL RADLW(ZLRZX(R),ZLRZY(R),ZHLZZ(R),ZKLZZ(R),ZULZZ(R),
     2               FQLWR(SIDE,W))
          END IF
  670   CONTINUE
  680 CONTINUE
      GO TO 10
C
C FQPWR:
  690 DO 720 W=1,NWALLS
        DO 710 SIDE=1,2
          IF (FFQPWR(SIDE,W).EQ.REGIM1) THEN
          R = WLINK(SIDE,W)
          SUM = 0.D0
          DO 700 O=1,OBJS(R)
            FQ1PW = 0.D0
            IF (STATE(O,R).EQ.FLAME)
     1       CALL RADFW(ZLRZX(R),ZLRZY(R),TEPZR(O,R),ZHLZZ(R),
     2          ZHPZZ(O,R),ZRFZZ(O,R),ZULZZ(R),FQ1PW)
            SUM = SUM + FQ1PW
  700     CONTINUE
          FQPWR(SIDE,W) = SUM
          END IF
  710   CONTINUE
  720 CONTINUE
      GO TO 10
C
C FQLWD:
  730 DO 750 W=1,NWALLS
        DO 740 SIDE=1,2
          IF (FFQLWD(SIDE,W).EQ.REGIM1) THEN
          R=WLINK(SIDE,W)
          IF (R.EQ.0) THEN
C USE THIS CALL IF THE WALL SIDE IS IN THE OUTSIDE WORLD.
            CALL CNVW1(ZOWZM,ZOWZN,ZKAOUT,ZKWZZ(SIDE,W),FQLWD(SIDE,W))
          ELSE
```

```
C USE THIS CALL IF WALL IS IN A ROOM.
            CALL CNVW1(ZOWZM,ZOWZN,ZKLZZ(R),ZKWZZ(SIDE,W),FCLWD(SIDE,W))
          END IF
          END IF
  740   CONTINUE
  750 CONTINUE
      GO TO 10
C
C ZKWZZ:
  760 DO 780 W=1,NWALLS
        DO 770 SIDE=1,2
          IF (FZKWZZ(SIDE,W).EQ.REGIM1) THEN
          R=WLINK(SIDE,W)
          IF (R.NE.0) THEN
            ZUL = ZULZZ(R)
            ZKA = ZKAZZ(R)
            ZHL = ZHLZZ(R)
          ELSE
            ZKA = ZKACUT
            ZHL = 0.D0
            ZUL = ZULCUT
          END IF
          CALL TMPW1 (H,SIDE,W,ZKA,FCLWD(SIDE,W),FQLWR(SIDE,W),
     1    FQFWR(SIDE,W),ZHL,ZKWZZ(SIDE,W),ZUL,ZKWZZ(SIDE,W))
          END IF
  770   CONTINUE
  780 CONTINUE
      GO TO 10
C
C TMODS:
  790 DO 810 R=1,NROOMS
        IF (FTMODS(R).EQ.REGIM1) THEN
        SUM = 0.D0
        DO 800 O=1,OBJS(R)
          TMOD1 = 0.D0
          IF (STATE(O,R).EQ.FLAME)
     1     CALL DXDPRD(FCO2(O,R),TMOZZ(O,R),TMOD1)
          SUM = SUM + TMOD1
  800   CONTINUE
        TMODS(R) = SUM
        END IF
  810 CONTINUE
      GO TO 10
C
C TMOMS:
  820 DO 840 R=1,NROOMS
        IF (FTMOMS(R).EQ.REGIM1) THEN
        SUM = 0.D0
        DO 830 O=1,OBJS(R)
          TMOM1 = 0.D0
          IF (STATE(O,R).EQ.FLAME)
     1     CALL MNXPRD(FCO(O,R),TMOZZ(O,R),TMOM1)
          SUM = SUM + TMOM1
  830   CONTINUE
        TMOMS(R) = SUM
        END IF
  840 CONTINUE
      GO TO 10
C
C TMOSS:
  850 DO 870 R=1,NROOMS
        IF (FTMOSS(R).EQ.REGIM1) THEN
        SUM = 0.D0
        DO 860 O=1,OBJS(R)
          TMOS1 = 0.D0
          IF (STATE(O,R).EQ.FLAME)
     1     CALL SUTPRD(FS(O,R),TMOZZ(O,R),TMOS1)
          SUM = SUM + TMOS1
  860   CONTINUE
        TMOSS(R) = SUM
        END IF
  870 CONTINUE
      GO TO 10
C
C TMOWS:
  880 DO 900 R=1,NROOMS
        IF (FTMOWS(R).EQ.REGIM1) THEN
        SUM = 0.D0
        DO 890 O=1,OBJS(R)
          TMOW1 = 0.D0
          IF (STATE(O,R).EQ.FLAME)
     1     CALL WTRPRD(FH2O(O,R),TMOZZ(O,R),TMOW1)
          SUM = SUM + TMOW1
  890   CONTINUE
        TMOWS(R) = SUM
        END IF
  900 CONTINUE
      GO TO 10
C
C TMOZB:
C NOTE THAT BFIR HAS NOT YET BEEN INCORPORATED.(9/3/81)
  910 DO 930 R=1,NROOMS
        DO 920 O=1,OBJS(R)
          IF (FTMOZB(O,R).EQ.REGIM1) THEN
          CALL BURN(CHI(O,R),XGAMMA(O,R),TMOZZ(O,R),TMPZZ(O,R),
     2              TMOZB(O,R))
          END IF
  920   CONTINUE
  930 CONTINUE
      GO TO 10
C
C PHI:
  940 DO 960 R=1,NROOMS
        DO 950 O=1,OBJS(R)
          IF (FPHI(O,R).EQ.REGIM1) THEN
          CALL FLUX(QVAP(O,R),ZUFZZ,FQLOR(O,R),FQPOR(O,R),
     2      FQWOR(O,R),ZMOZZ(O,R),ZRFZZ(O,R),PHI(O,R))
          END IF
  950   CONTINUE
  960 CONTINUE
      GO TO 10
C
C AVLSM:
  970 DO 990 R=1,NROOMS
        IF (FAVLSM(R).EQ.REGIM1) THEN
        SUM = 0.D0
        DO 980 V=1,NVENTS
          AVL1 = 0.D0
          IF (VLINK(1,V).EQ.R .OR. VLINK(2,V).EQ.R)
     1    CALL VNTCVR(ZBVZZ(V),ZHTZZ(V),ZHVZZ(V),ZHLZZ(R),AVL1)
          SUM = SUM + AVL1
  980   CONTINUE
        AVLSM(R) = SUM
        END IF
  990 CONTINUE
      GO TO 10
```

```
C
C WVLS:
 1000 DO 1020 R=1,NROOMS
        IF (FWVLS(R).EQ.REGIM1) THEN
        SUM = 0.D0
        DO 1010 V=1,NVENTS
          WVL1 = 0.D0
          IF (VLINK(1,V).EQ.R .OR. VLINK(2,V).EQ.R)
     1    CALL CVWID(ZBVZZ(V),ZHTZZ(V),ZHVZZ(V),ZHLZZ(R),WVL1)
          SUM = SUM + WVL1
 1010   CONTINUE
        WVLS(R) = SUM
        END IF
 1020 CONTINUE
      GO TO 10
C
C PPLR:
 1030 DO 1050 R=1,NROOMS
        IF (FPPLR(R).EQ.REGIM1) THEN
        SUM = 0.D0
        DO 1040 O=1,OBJS(R)
          P1PLR1 = 0.D0
          IF (STATE(O,R).EQ.FLAME)
     1    CALL PWRPLR(ZLRZX(R),ZLRZY(R),TEPZR(O,R),ZHPZZ(O,R),
     2      ZRFZZ(O,R),P1PLR1)
          SUM = SUM + P1PLR1
 1040   CONTINUE
        PPLR(R) = SUM
        END IF
 1050 CONTINUE
      GO TO 10
C
C TMOSM:
 1060 DO 1080 R=1,NROOMS
        IF (FTMOSM(R).EQ.REGIM1) THEN
        SUM = 0.D0
        DO 1070 O=1,OBJS(R)
          IF (STATE(O,R).EQ.FLAME) SUM = SUM + TMOZZ(O,R)
 1070   CONTINUE
        TMOSM(R) = SUM
        END IF
 1080 CONTINUE
      GO TO 10
C
C TMPSM:
 1090 DO 1110 R=1,NROOMS
        IF (FTMPSM(R).EQ.REGIM1) THEN
        SUM = 0.D0
        DO 1100 O=1,OBJS(R)
          IF (STATE(O,R).EQ.FLAME) SUM = SUM + TMPZZ(O,R)
 1100   CONTINUE
        TMPSM(R) = SUM
        END IF
 1110 CONTINUE
      GO TO 10
C
C FLOWS (TEUSM,TMOSM,TMUSM,TMOZF,TMDZF,TMMZF,TMSZF,TMWZF):
 1120 DO 1130 R=1,NROOMS
        DO 1125 O=1,8
          SUM1(O,R) = 0.D0
 1125   CONTINUE
 1130 CONTINUE
C
```

```
      DO 1150 V=1,NVENTS
        TEU = 0.D0
        TMU = 0.D0
        TMD = 0.D0
        R1 = VLINK(1,V)
        R2 = VLINK(2,V)
        IF (R1.EQ.0) THEN
          CALL FLOW(CD,CP,(VMAOUT+VMAZZ(R1))/2.D0,ZBVZZ(V),ZHRZZ(R2),
     1     ZHTZZ(V),ZHVZZ(V),ZKAOUT,ZKAZZ(R2),0.D0,ZHLZZ(R2),ZKAOUT,
     2     ZKLZZ(R2),0.D0,ZPRZZ(R2),TEU,TMD,TMU)
        ELSE IF (R2.EQ.0) THEN
          CALL FLOW(CD,CP,(VMAOUT+VMAZZ(R1))/2.D0,ZBVZZ(V),ZHRZZ(R1),
     1     ZHTZZ(V),ZHVZZ(V),ZKAZZ(R1),ZKAOUT,ZHLZZ(R1),0.D0,ZKLZZ(R1),
     2     ZKAOUT,ZPRZZ(R1),0.D0,TEU,TMD,TMU)
        ELSE
          IF (ZHRZZ(R1).NE.ZHRZZ(R2)) WRITE(IWTTY,1140)
 1140     FORMAT(' WARNING--FLOW ASSUMES THAT ADJOINING ROOMS HAVE THE
     1SAME HEIGHT.')
          CALL FLOW(CD,CP,(VMAZZ(R1)+VMAZZ(R2))/2.D0,ZBVZZ(V),ZHRZZ(R1),
     1     ZHTZZ(V),ZHVZZ(V),ZKAZZ(R1),ZKAZZ(R2),ZHLZZ(R1),ZHLZZ(R2),
     2     ZKLZZ(R1),ZKLZZ(R2),ZPRZZ(R1),ZPRZZ(R2),TEU,TMD,TMU)
        END IF
C
        IF (R1.NE.0) THEN
          SUM1(1,R1) = SUM1(1,R1) + TEU
          SUM1(2,R1) = SUM1(2,R1) + TMD
          SUM1(3,R1) = SUM1(3,R1) + TMU
        END IF
        IF (R2.NE.0) THEN
          SUM1(1,R2) = SUM1(1,R2) - TEU
          SUM1(2,R2) = SUM1(2,R2) - TMD
          SUM1(3,R2) = SUM1(3,R2) - TMU
        END IF
C
C GET THE MASS FLOW OF THE VARIOUS GASEOUS SPECIOUS THROUGH THE VENT.
        IF (TMU .GE. 0.D0) THEN
          R = R1
        ELSE
          R = R2
        END IF
        IF (R .NE. 0) THEN
          CALL SPECFL(TMU,ZYLDZ(R),ZYLWZ(R),ZYLSZ(R),ZYLMZ(R),
     1 ZYLOZ(R),CO2FLO,H2OFLO,SUTFLO,MNXFLO,OXYFLO)
        ELSE
          CALL SPECFL(TMU,YCO2,0.D0,0.D0,0.D0,YO2,
     1 CO2FLO,H2OFLO,SUTFLO,MNXFLO,OXYFLO)
        END IF
        IF (R1.NE.0) THEN
          SUM1(4,R1) = SUM1(4,R1) + CO2FLO
          SUM1(5,R1) = SUM1(5,R1) + H2OFLO
          SUM1(6,R1) = SUM1(6,R1) + SUTFLO
          SUM1(7,R1) = SUM1(7,R1) + MNXFLO
          SUM1(8,R1) = SUM1(8,R1) + OXYFLO
        END IF
        IF (R2.NE.0) THEN
          SUM1(4,R2) = SUM1(4,R2) - CO2FLO
          SUM1(5,R2) = SUM1(5,R2) - H2OFLO
          SUM1(6,R2) = SUM1(6,R2) - SUTFLO
          SUM1(7,R2) = SUM1(7,R2) - MNXFLO
          SUM1(8,R2) = SUM1(8,R2) - OXYFLO
        END IF
 1150 CONTINUE
C
```

```
      DO 1160 R=1,NROOMS
        IF (FTEUSM(R).EQ.REGIM1) TEUSM(R) = SUM1(1,R)
        IF (FTMDSM(R).EQ.REGIM1) TMDSM(R) = SUM1(2,R)
        IF (FTMUSM(R).EQ.REGIM1) TMUSM(R) = SUM1(3,R)
        IF (FTMDZF(R).EQ.REGIM1) TMDZF(R) = SUM1(4,R)
        IF (FTMWZF(R).EQ.REGIM1) TMWZF(R) = SUM1(5,R)
        IF (FTMSZF(R).EQ.REGIM1) TMSZF(R) = SUM1(6,R)
        IF (FTMMZF(R).EQ.REGIM1) TMMZF(R) = SUM1(7,R)
        IF (FTMOZF(R).EQ.REGIM1) TMOZF(R) = SUM1(8,R)
 1160 CONTINUE
      GO TO 10
C
C EPLSM:
 1170 DO 1190 R=1,NROOMS
        IF (FEPLSM(R).EQ.REGIM1) THEN
        SUM = 0.D0
        DO 1180 O=1,OBJS(R)
          TMPLU1 = 0.D0
          IF (STATE(O,R).EQ.FLAME)
     1    CALL ENTPL(TMPLU1)
          SUM = SUM + TMPLU1
 1180   CONTINUE
        EPLSM(R) = SUM
        END IF
 1190 CONTINUE
      GO TO 10
C
C BOXYS:
 1200 DO 1220 R=1,NROOMS
        IF (FBOXYS(R).EQ.REGIM1) THEN
        SUM = 0.D0
        DO 1210 O=1,OBJS(R)
          BOXY1 = 0.D0
          IF (STATE(O,R) .EQ. FLAME)
     1     CALL OXYBRN(XGAMAS(O,R),TMOZB(O,R),BOXY1)
          SUM = SUM + BOXY1
 1210   CONTINUE
        BOXYS(R) = SUM
        END IF
 1220 CONTINUE
      GO TO 10
C
C
C TEPSM:
 1230 DO 1250 R=1,NROOMS
        IF (FTEPSM(R).EQ.REGIM1) THEN
        SUM = 0.D0
        DO 1240 O=1,OBJS(R)
          TEPZZ = 0.D0
          IF (STATE(O,R).EQ.FLAME)
     1    CALL TENPLM(ZKAZZ(R),ZKCPY(O,R),TECZZ(O,R),TEPZR(O,R),
     2                TMOZZ(O,R),TMPZZ(O,R),TEPZZ)
          SUM = SUM + TEPZZ
 1240   CONTINUE
        TEPSM(R) = SUM
        END IF
 1250 CONTINUE
      GO TO 10
C
C
C TEOSM:
 1260 DO 1280 R=1,NROOMS
        IF (FTEOSM(R).EQ.REGIM1) THEN
```

```
         SUM = 0.D0
         DO 1270 O=1,OBJS(R)
           IF (STATE(O,R) .EQ. FLAME) SUM = SUM + TEOZZ(O,R)
 1270    CONTINUE
         TEOSM(R) = SUM
         END IF
 1280 CONTINUE
      GO TO 10
      END
      SUBROUTINE FUNCT(X,ZERO,TIME)
      IMPLICIT INTEGER (A-Z)
C
C+        PART OF MODULE INTERFACE
      REAL*8 X,ZERO,TIME
      DIMENSION X(*),ZERO(*)
C
C ALDIFF CALLS THIS SUBROUTINE TO EVALUATE THE PHYSICAL EQUATIONS.
C FUNCT PERFORMS THREE OPERATIONS:
C            STEP 1 - FUNCT UNPACKS THE INPUT VECTOR X (NOT THE SAME
C            X AS USED IN THE EQS. ABOVE) INTO SECTION 1 OF /VAR/.
C            EVERY VARIABLE CURRENTLY IN THE SYSTEM IS ASSIGNED IN
C            THIS UNPACKING.  THE VARIABLES WHICH ARE NOT IN THE
C            SYSTEM ARE INITIALLY GIVEN MEANINGFUL VALUES IN
C            SETSYS AND RETAIN
C            THESE VALUES.  FOR EXAMPLE, TMOZZ OF A COOL OBJECT IS
C            INITIALLY SET TO 0. AND IS NOT REASSIGNED UNTIL THAT
C            OBJECT IGNITES.  THUS, AFTER THE UNPACKING STEP, EVERY
C            PHYSICAL VARIABLE IN SECTION 1 OF /VAR/ CONTAINS A
C            PHYSICALLY MEANINGFUL VALUE.
C
C            STEP 2 - FUNCT CALLS EVALFP AND ERFDE TO EVALUATE THE
C            VECTOR VALUED FUNCTIONS F, G, H, I, WHERE F REPRESENTS
C            THE ROOT FINDING EQUATIONS, G REPRESENTS THE DIFFERENTIAL
C            EQUATIONS, H REPRESENTS THE FIXED POINT EQUATIONS, AND I
C            REPRESENTS THE ELIMINATED VARIABLE EQUATIONS.
C            EVALFP (EVALUATE FIXED POINT) CALLS THE FUNCTIONS
C            ASSOCIATED WITH THE FIXED POINT EQUATIONS, THE COMPONENTS
C            OF H AND I, AND PLACES THE OUTPUT INTO THE FIRST SECTION
C            OF /VAR/.  ERFDE (EVALUATE ROOT FINDING AND DIFFERENTIAL
C            EQUATIONS) CALLS THE ROOT FINDING AND DIFFERENTIAL
C            FUNCTIONS, THE COMPONENTS OF F AND G RESPECTIVELY AND
C            PLACES THE OUTPUT IN SECTIONS 2 AND 3 OF /VAR/
C            RESPECTIVELY.
C
C            STEP 3 - THE CONTENTS OF /VAR/ IS REPACKED INTO THE
C            OUTPUT ARRAY ZERO AS DIRECTED BY THE POINTER ARRAY
C            PACK.
C
C NOTE THAT THE VARIABLES ARE SEQUENTIALLY UNPACKED FROM X AND PLACED
C INTO /VAR/ IN THE POSITION INDICATED BY THE CORRESPONDING ELEMENT OF
C THE ARRAY INSYS.  SIMILARLY THE ARRAY PACK DETERMINES THE ORDER IN
C WHICH VARIABLES ARE PACKED INTO THE OUTPUT ARRAY ZERO.
C
C NOTE ON ORDERING AND VARIABLE ELIMINATION:
C      WHEN EVALUATING THE FIXED
C POINT FUNCTIONS (THAT IS THE FUNCTIONS ASSOCIATED WITH THE FIXED
C POINT EQUATIONS), EVALFP SENDS INPUT VARIABLES TO THE PHYSICAL
C SUBROUTINES FROM SECTION 1 OF /VAR/ AND ALSO PLACES EACH OUTPUT
C FROM THE PHYSICAL SUBROUTINES INTO SECTION 1 OF /VAR/.  SINCE THE
C INPUTS TO THE N'TH PHYSICAL SUBROUTINE ARE AFFECTED BY THE OUTPUTS
C OF THE N-1, N-2, ETC. SUBROUTINES, THE CALLS IN EVALFP CAN BE
C ORDERED TO GIVE A MINIMUM NUMBER OF FIXED POINT VARIABLES (I.E. THE
C ORDERING CAN BE USED TO ACHEIVE VARIABLE ELIMINATION.)
```

```
C      THE EVALUATION ORDER OF THE ROOT FINDING AND DIFFERENTIAL
C EQUATIONS IS LESS IMPORTANT BUT MAY STILL BE SIGNIFICANT.  WHEN
C EVALUATING THE FUNCTION ASSOCIATED WITH A ROOT-FINDING OR DIFFEREN-
C TIAL EQUATION, ERFDE ALSO SENDS INPUT VARIABLES FROM SECTION 1 OF
C /VAR/ TO THE PHYSICAL SUBROUTINES.  HOWEVER, ERFDE PLACES THE OUTPUTS
C RETURNED BY THE PHYSICAL SUBROUTINES INTO SECTIONS 2 AND 3 OF /VAR/.
C THUS, THE OUTPUT OF THE N'TH ROOT-FINDING OR DIFFERENTIAL FUNCTION
C CANNOT AFFECT THE INPUTS TO THE N+1 SUCH FUNCTION.  THEREFORE, THE
C EVALUATION ORDER OF THE ROOT-FINDING AND DIFFERENTIAL FUNCTCINS WITH
C EACH OTHER IS NOT IMPORTANT.
C      IF THE EVALUATION OF A ROOT FINDING OR DIFFERENTIAL FUNCTION
C IS REORDERED WITH ANY FIXED POINT FUNCTION EVALUATION, HOWEVER,
C THEN THE INPUTS TO THE ROOT-FINDING OR DIFFERENTIAL FUNCTION MAY
C CHANGE.  THE TWO ORDERS OF EVALUATION SHOULD PRODUCE SIMILAR
C SOLUTIONS TO THE NUMERICAL PROBLEM, BUT ONE ORDER OF EVALUATION MAY
C BE MORE STABLE THAN ANOTHER.
C
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,VAR6.CMN
      INTEGER I
      REAL*8 XIN(VARSIZ)
C NOTE THAT THE FOLLOWING EQUIVALENCE ASSUMES THAT TELZR IS THE FIRST
C ARRAY IN COMMON /VAR/
      EQUIVALENCE (XIN,TELZR)
      INTEGER PACK
      COMMON /PCKPTR/ PACK(MAXVRS+1)
C
C UNPACK X INTO COMMON /VAR/.
      CALL UNPACK(X)
C
C PERFORM FUNCTION EVALUATION.
C CURRENTLY, THE ROOT FINDING AND DIFFERENTIAL EQUATIONS ARE EVALUATED
C AFTER ALL OF THE FIXED POINT EQUATIONS.
      CALL EVALFP(TIME)
      CALL ERFDE(TIME)
C
C REPACK INTO ZERO.
      I = 1
C    WHILE PACK(I) <>0 DO
   10 IF (PACK(I) .EQ. 0) GO TO 20
        ZERO(I) = XIN(PACK(I))
        I = I + 1
        GO TO 10
C     END WHILE
   20 CONTINUE
C
      RETURN
      END
      PROGRAM MARK6
      IMPLICIT INTEGER (A-Z)
C
C  This is the main controlling program of version 6 of the Home Fire
C  Code.  The structure of this module is given below.
C
C      initializations (including call to the input package)
C      while time < finish do
C        check for changes in set of equations
C        check for outputs
C        locate next planned interruption - set TINT to time of interruption
C        while time < tint do
C          set up for a time step - get size of time step to try
C          call integrator
C          if step failed, reset to first order integration or bomb out
```

```
C         see if any unplanned discontinuities occurred and reject step if yes
C       end while
C     end while
C     stop
C
C  Note that if an interruption is scheduled to occur so that the smallest
C  allowable time step will step beyond the time of the interruption, then
C  the interruption is caused to occur at the beginning of the time step.
C  In other words, every time the subroutine "checks to see if it's time to
C  do something", it checks via IF (TIME+HMIN.GT.time to do something) THEN.
C
C
C  Local Variables:
C     NT -- the number of the time step
C     NXTTY -- time for the next output to the
C     NXTDSK -- time for the next output to the disk file
C     TDISC -- time of the next known discontinuity or change in the set
C              of equations; set to 2*FINISH if there is no scheduled
C              discontinuity
C     TINT -- time of the next known interrruption
C     TIME -- time
C     OLDTIM -- time at which converged values were stored in OLDX
C     X -- values for the variables and their time derivatives
C     XMAX -- contains a significant value for each of the variables in
C             the system
C     H -- size of the time step
C     HMIN -- smallest allowable time step
C     HMAX -- largest allowable time step
C     EPS -- the one step error tolerance
C     OLDX -- the converged values of the variables and their time
C             derivatives at the previous time step
C     ERROR -- the estimates of the one step truncation error for
C              each of the variables
C     ZERO,Z,JAC,PIVOT -- work space for the numerics
C     HSTORE -- the size of the time step to which the derivatives in
C               X are scaled
C     HOLDST -- the size of the time step to which the derivatives in
C               OLDX are scaled
C     HSTART -- the size of the time step that will be attempted after
C               each restart of the integrator
C     FINISH -- stopping time for the run
C     TTYOUT -- the output interval for the terminal in seconds
C     DSKOUT -- the output interval for the disk file in seconds
C     PIVOT -- used by the numerics
C     RESULT -- returned by ALDIFF; gives information on the step
C               just completed; see ALDIFF comments for details
C     ORDER -- the current order of the method
C     MAXODR -- the maximum allowable order of the method
C     OLDORD -- the order of the method before a last call to ALDIFF
C     DSCTYP -- returned by UNPDSC; the type of discontinuity which has
C               been detected
C     CHNGRM -- the room in which the above discontinuity has been
C               detected
C     CHNGOB -- the object which has experienced the above discontinuity
C     FNUM -- number of root finding equations in the system
C     GNUM -- number of differential equations in the system
C     HNUM -- number of fixed point equations in the system
C     INUM -- number of eliminated variable equations in the system
C     SIZE -- number of equations and variables in the system
C
      REAL*8 NXTTY,NXTDSK,TDISC,TINT,TIME,OLDTIM
      REAL*8 X,XMAX,H,HMIN,HMAX,EPS,OLDX,ERROR,ZERO,Z,JAC,HOLD,
     1    HSTORE,HOLDST,HSTART,FINISH,TTYOUT,DSKOUT
```

```
      INTEGER PIVOT,RESULT,ORDER,MAXODR,J,I,OLDORD,DSCTYP,CHNGRM,
     1 CHNGOB,FNUM,GNUM,HNUM,INUM,SIZE
      CHARACTER VARNAM*10
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,IO6.CMN
$INCLUDE 9,DSCNAM.CMN
      PARAMETER (MAXODR=4,NDIM = 100)
      DIMENSION X(MAXVRS*(MAXODR+1)),XMAX(MAXVRS),
     1  OLDX(MAXVRS*(MAXODR+1)),
     2  ERROR(2*MAXVRS),ZERO(MAXVRS),Z(MAXVRS),JAC(MAXVRS,NDIM),
     3  PIVOT(NDIM)
      DATA HSTART/1.D-2/
      DATA EPS,HMIN,HMAX/1.D-3,1.D-4,5.D0/
C
C CALL TO INITIALIZE THE RUN TIMER.  FOR USE ON HARVARD VAX 11-780
C COMPUTERS ONLY.
      CALL RTIME(0)
      OPEN (UNIT=IWDSK,FILE='FIRE.DAT',STATUS='NEW',
     1  ACCESS='SEQUENTIAL',RECL=150)
      WRITE(IWDSK,10) HMIN,HMAX,EPS,MAXODR
      WRITE(IWTTY,10) HMIN,HMAX,EPS,MAXODR
   10 FORMAT(' HMIN=',1PE10.3,3X,'HMAX=',1PE10.3,3X,'EPS=',1PE10.3,
     1     3X,'MAXODR=',I3)
      TIME = 0.D0
      TDISC = 0.D0
      DSCTYP = STRTUP
      NXTTY = 0.D0
      NXTDSK = 0.D0
      NT = 0
      CALL INIT(FINISH,TTYOUT,DSKOUT)
      CALL NEWSTP(HMAX,TIME,X)
C
C WHILE TIME + HMIN <= FINISH DO
   20 IF (TIME + HMIN .GT. FINISH) GO TO 100
C
C CHECK FOR CHANGES IN THE SET OF EQUATIONS.
      IF (TIME + HMIN .GT. TDISC) THEN
        H = HSTART
        CALL NWSTAT(DSCTYP,CHNGOB,CHNGRM,X,FNUM,GNUM,HNUM,INUM,SIZE,
     1   ZERO,Z,TIME,EPS,XMAX,PIVOT,JAC,MAXVRS)
        HSTORE = H
        CALL ADINIT(X,TIME,HSTORE,EPS,ORDER,FNUM,GNUM,HNUM,INUM)
        TDISC = 2.D0*FINISH
      END IF
C
C CHECK FOR OUTPUT TO THE DISK.
      IF (TIME + HMIN .GT. NXTDSK) THEN
        CALL WRIT(X,TIME,NT,IWDSK)
        NXTDSK = NXTDSK+DSKOUT
      END IF
C
C CHECK FOR OUTPUT TO THE TERMINAL SCREEN.
      IF (TIME + HMIN .GT. NXTTY) THEN
        CALL WRIT(X,TIME,NT,IWTTY)
        NXTTY = NXTTY+TTYOUT
      END IF
C
C LOCATE NEXT PLANNED INTERRUPTION:
      TINT = MIN(FINISH,NXTTY,NXTDSK)
C
C INTEGRATE UP TO A DISCONTINUITY:
C WHILE (TIME + HMIN .LE. TINT) DO
   30 IF (TIME + HMIN .GT. TINT) GO TO 110
```

```
C
C BEGIN NEW TIME STEP.
      CALL NEWSTP(HMAX,TIME,X)
      CALL SCALE(X,XMAX)
C
C SAVE ENOUGH INFORMATION SO THAT WE CAN RECOVER IF WE
C STEP PAST AN UNPLANNED DISCONTINUITY.
      OLDTIM = TIME
      OLDORD = ORDER
      HOLD = H
      HOLDST = HSTORE
C
C TAKE A STEP.
C  SET UP THE TIME STEP SO AS NOT TO STEP PAST A DISCONTINUITY OR OTHER
C  TYPE OF INTERRUPTION.
      H = MIN(H,TINT-TIME)
      H = MIN(HMAX,H)
      H = MAX(HMIN,H)
      CALL ALDIFF(X,XMAX,TIME,H,HSTORE,HMIN,EPS,
     1   RESULT,ORDER,MAXVRS,MAXODR,
     1     OLDX,ERROR,ZERO,Z,JAC,PIVCT)
C
C WRITE OUT INFORMATION ON THE STEP JUST TAKEN.
C
      WRITE (IWDSK,40) TIME,H,RESULT,ORDER
      WRITE (IWTTY,40) TIME,H,RESULT,ORDER
   40 FORMAT(7X,'T=',F7.3,'   H=',1PE10.3,'   RESULT=',I5,
     1 '   ORDER=',I3)
C
C IF THE STEP FAILED, TAKE APPROPRIATE ACTION.
C
      IF (RESULT .LT. 0) THEN
C ALDIFF FAILED.
        IF (ORDER .NE. 1) THEN
          H = HMIN
          HSTORE = H
          CALL ADINIT(X,TIME,HSTORE,EPS,ORDER,FNUM,GNUM,HNUM,INUM)
        ELSE
C
C COME HERE IF THE PROGRAM BOMBED OUT.  WRITE OUT ERROR ARRAY AND
C THE LAST CONVERGED SET OF VALUES THEN QUIT.
          DO 60 I = 1,SIZE
            ERROR(I) = ERROR(I)/XMAX(I)
   60     CONTINUE
C         WRITE(IWTTY,70) (VARNAM(I),ERROR(I),I=1,SIZE)
C         WRITE(IWDSK,70) (VARNAM(I),ERROR(I),I=1,SIZE)
C  70     FORMAT(X,'SCALED TRUNCATION ERRORS'/
C    1<SIZE/3+1>(X,3(:,A10,'=',1PE10.3,5X)/))
          WRITE(IWTTY,69)
   69     FORMAT(' SCALE TRUNCATION ERRORS')
          WRITE(IWTTY,70) (VARNAM(I),ERROR(I),I=1,SIZE)
   70     FORMAT(1X,(3(A10,'=',1PE10.3,5X),/,1X))
          WRITE(IWDSK,69)
          WRITE(IWDSK,70) (VARNAM(I),ERROR(I),I=1,SIZE)
C
C         WRITE(IWTTY,80) (VARNAM(I),OLDX(I),I=1,SIZE)
C         WRITE(IWDSK,80) (VARNAM(I),OLDX(I),I=1,SIZE)
C  80     FORMAT(X/X,'LAST CONVERGED VALUES:'/
C    1<SIZE/3+1>(X,3(:,A10,'=',1PE10.3,5X)/))
          WRITE(IWTTY,79)
   79     FORMAT(' LAST CONVERGED VALUES :')
          WRITE(IWTTY,70) (VARNAM(I),OLDX(I),I=1,SIZE)
          WRITE(IWDSK,79)
```

```
            WRITE(IWDSK,70) (VARNAM(I),OLDX(I),I=1,SIZE)
C
C WRITE OUT CPU TIME BEFORE STOPPING.  FOR USE ON HARVARD VAX 11-780
C COMPUTER ONLY.
            CALL RTIME(1)
            STOP
          END IF
        ELSE
C ALDIFF SUCCEEDED.
C
C SEE IF ANY UNPLANNED DISCONTINUITIES OCCURRED.
C
          CALL UNPDSC(X,TIME,TDISC,FINISH,DSCTYP,CHNGOB,CHNGRM)
C
C IF AN UNPLANNED DISCONTINUITY OCCURRED, WE MUST BACK UP AND REPEAT
C THE TIME STEP, BEING CAREFUL NOT TO STEP BEYOND THE ESTIMATED TIME
C OF THE DISCONTINUITY.
          IF (TDISC .LE. FINISH) TINT = TDISC
          IF (TDISC .LT. TIME) THEN
            DO 90 I = 1, SIZE*(OLDORD+1)
              X(I) = OLDX(I)
   90       CONTINUE
            TIME = OLDTIM
            ORDER = OLDORD
            H = HOLD
            HSTORE = HOLDST
            TINT = TDISC
          END IF
        END IF
        NT = NT + 1
C
C END WHILE
        GO TO 30
  110   CONTINUE
C END WHILE
        GO TO 20
  100   CONTINUE
C
        CALL WRIT(X,TIME,NT,IWTTY)
        CALL WRIT(X,TIME,NT,IWDSK)
        CALL RTIME(1)
        STOP
        END
        SUBROUTINE INIT(FINISH,TTYOUT,DSKOUT)
        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 FINISH,TTYOUT,DSKOUT
C
C THIS SUBROUTINE IS CALLED AT THE START OF A RUN TO:
C  1) READ THE DATA FILE CREATED BY THE INPUT PROGRAM TO GET INITIAL
C     VALUES FOR THE PHYSICAL PARAMETERS IN THE RUN.
C  2) LABEL THE PRINTED OUTPUT WITH USER COMMENTS, DATE AND TIME, RUN
C     NUMBER CODE, AND INPUT PARAMETERS:
$INCLUDE 9,IO6.CMN
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,BLDNG.CMN
        INTEGER INPSIZ
        PARAMETER (INPSIZ=5380/4)
        INTEGER IDAT(INPSIZ)
        EQUIVALENCE (IDAT,ZBVZZ)
        CHARACTER PROSE*80
C
    5 WRITE(IWTTY,10)
   10 FORMAT(' ENTER THE NAME OF THE FILE WHICH CONTAINS THE DATA FOR
     1 THIS RUN.')
       READ(IRTTY,20,ERR=5) PROSE(1:30)
    20 FORMAT(A)
       OPEN(UNIT=21,ACCESS='SEQUENTIAL',FORM='UNFORMATTED',STATUS='OLD',
     1 FILE=PROSE(1:30))
       DO 15 I=1,INPSIZ
       READ(21) IDAT(I)
    15 CONTINUE
       CALL RECAP(IWDSK)
       WRITE(IWDSK,30) VERSUN(1),VERSUN(2)
    30 FORMAT(X,'VERSUN(1)=',I2,'  VERSUN(2)=',I2)
       WRITE(IWDSK,40) (I,VLINK(1,I),VLINK(2,I),I=1,NVENTS)
    40 FORMAT(' VENT ',I2,': SIDE 1 IN ROOM ',I2,' SIDE 2 IN ROOM ',I2)
    80 WRITE (IWTTY,350) 'TTY OUTPUT INTERVAL','20.0'
   350 FORMAT(' ENTER ',A23,' DEFAULT IS ',A7,': ')
       READ (IRTTY,240,ERR=80) TTYOUT
   240 FORMAT(E12.5)
       IF (TTYOUT .LE. 0.D0) TTYOUT = 20.D0
    90 WRITE (IWTTY,350) 'DISK OUTPUT INTERVAL','10.0'
       READ (IRTTY,240,ERR=90) DSKOUT
       IF (DSKOUT .LE. 0.D0) DSKOUT = 10.D0
C
C ASK USER FOR LOOP CONTROL PARAMS, OR DEFAULT
  100 WRITE (IWTTY,350) 'RUN LENGTH     ','500.0'
       READ (IRTTY,240,ERR=100) FINISH
       IF (FINISH .LE. 0.D0) FINISH = 500.D0
       RETURN
       END
       SUBROUTINE NWSTAT(DSCTYP,OB,RM,X,FNUM,GNUM,HNUM,INUM,SIZE,
     1 ZERO,Z,TIME,EPS,XMAX,PIVOT,JAC,MAXVR1)
       IMPLICIT INTEGER (A-Z)
       INTEGER DSCTYP,OB,RM,FNUM,GNUM,HNUM,INUM,SIZE,PIVOT(*),
     1   MAXVR1
       REAL*8 X(*),ZERO(*),Z(*),TIME,EPS,XMAX(*),JAC(MAXVR1,*)
C
C MARK6 CALLS THIS SUBROUTINE WHENEVER THERE IS A CHANGE IN
C THE SET OF EQUATIONS OR A DISCONTINUITY IN THE VARIABLES.
C THE SUBROUTINE SETS UP THE VALUES IN THE ARRAYS STATE,
C FIRTYP, LAYR, AND FIRCOM TO REFLECT THE STATE CHANGE (IF ANY)
C WHICH HAS OCCURRED.  THE SUBROUTINE THEN CALLS SETSYS
C TO SET UP THE POINTER ARRAYS INSYS AND PACK AND TO GET
C A DESCRIPTION OF THE NUMBER AND TYPE OF EACH EQUATION
C IN THE SYSTEM.  FINALLY, THE SUBROUTINE TRIES TO GET A
C CONVERGED SET OF VALUES FOR THE VARIABLES AFTER THE DISCON-
C TINUITY BY MAKING A GOOD INITIAL GUESS AND THEN CALLING THE
C NUMERICS TO SOLVE THE NEW SET OF SIMULTANEOUS EQUATIONS.
C
C INPUTS:
C     DSCTYP -- THE TYPE OF DISCONTINUITY WHICH HAS OCCURRED:
C               DSCTYP = 1 (BRNOUT) --> A BURNOUT.
C               DSCTYP = 2 (HEATIG) --> A HEAT IGNITION.
C               DSCTYP = 3 (CNTCIG) --> A CONTACT IGNITION.
C               DSCTYP = 4 (STRTUP) --> THE RUN HAS JUST STARTED.
C               DSCTYP = 5 (SPLOVR) --> HOT GASES BEGIN TO FLOW INTO
C                     A PREVIOUSLY UNINVOLVED ROOM.
C     OB -- THE OBJECT NUMBER OF THE OBJECT WHICH HAS CHANGED STATE
C           (IF APPLICABLE).
C     RM -- THE ROOM NUMBER OF THE ROOM IN WHICH THE OBJECT THAT
C           HAS CHANGED STATE IS LOCATED (IF APPLICABLE).
C     X -- CONTAINS THE SET OF CONVERGED VALUES FOR THE VARIABLES
C          IN THE SYSTEM BEFORE THE DISCONTINUITY.
C     TIME -- THE TIME AT WHICH THE DISCONTINUITY OCCURRED.
C     EPS -- THE ONE STEP ERROR PARAMETER.
```

```
C
C NOTE THAT THE FOLLOWING ARRAYS MUST BE PASSED AS INPUT TO BE
C USED AS WORK SPACE THOUGH THE INPUT AND OUTPUT CONTENTS ARE
C UNIMPORTANT.
C    ZERO(MAXVR1),Z(MAXVR1),XMAX(MAXVR1),PIVOT(MAXVR1),JAC(MAXVR1,NDIM)
C
C
C OUTPUT:
C      X -- CONTAINS THE SET OF CONVERGED VALUES FOR THE VARIABLES IN
C           THE SYSTEM AFTER THE DISCONTINUITY.
C      FNUM -- THE NUMBER OF ROOT FINDING EQUATIONS IN THE SYSTEM AFTER
C              THE DISCONTINUITY.
C      GNUM -- THE NUMBER OF DIFFERENTIAL EQUATIONS IN THE SYSTEM AFTER
C              THE DISCONTINUITY.
C      HNUM -- THE NUMBER OF FIXED POINT EQUATIONS IN THE SYSTEM AFTER THE
C              DISCONTINUITY.
C      INUM -- THE NUMBER OF ELIMINATED VARIABLE EQUATIONS IN THE SYSTEM
C              AFTER THE DISCONTINUITY.
C      SIZE -- THE NUMBER OF EQUATIONS IN THE SYSTEM AFTER THE
C              DISCONTINUITY.
C
      SAVE
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,BLDNG.CMN
$INCLUDE 9,VAR6.CMN
$INCLUDE 9,IC6.CMN
$INCLUDE 9,DSCNAM.CMN
      INTEGER O,R,W,SIDE
      LOGICAL BLAZE,SING,CONV
      REAL*8 PI,ZKA
      PARAMETER (PI = 3.1415927)
C
C THE FOLLOWING LINE UNPACKS THE VECTOR X OF CONVERGED VALUES INTO
C COMMON /VAR/.  AFTER THIS CALL, EVERY VARIABLE IN /VAR/ WHICH
C WAS DEFINED BEFORE THE DISCONTINUITY CONTAINS THE MEANINGFUL
C CONVERGED VALUE FROM JUST BEFORE THE TIME OF THE DISCONTINUITY.
      CALL UNPACK(X)
C
C THE FOLLOWING IF...ELSE IF...END IF CONSTRUCTION CHANGES THE VALUES
C IN THE ARRAYS STATE,FIRTYP,LAYR,AND FIRCOM SO THAT THEY DESCRIBE THE NEW
C STATE OF THE PROBLEM, THAT IS THE STATE AFTER THE DISCONTINUITY.
      IF (DSCTYP.EQ.SPLOVR) THEN
        LAYR(RM) = .TRUE.
      ELSE IF (DSCTYP .EQ. BRNOUT) THEN
        STATE(OB,RM) = CNSUM
        BLAZE = .FALSE.
        DO 10 O=1,OBJS(RM)
          BLAZE = BLAZE .OR. STATE(O,RM).EQ.FLAME
   10   CONTINUE
        FIROCM(RM) = BLAZE
C
      ELSE IF (DSCTYP .EQ. HEATIG) THEN
        ZTIG(OB,RM) = TIME
        STATE(OB,RM) = FLAME
        FIRTYP(OB,RM) = PCOL
        FIROCM(RM) = .TRUE.
        LAYR(RM) = .TRUE.
C
      ELSE IF (DSCTYP .EQ. CNTCIG) THEN
        ZTIG(OB,RM) = TIME
        STATE(OB,RM) = FLAME
        FIRTYP(OB,RM) = GROWNG
        FIROCM(RM) = .TRUE.
        LAYR(RM) = .TRUE.
C
C DISCONTINUITY IS THE START OF THE RUN
      ELSE IF (DSCTYP .EQ. STRTUP) THEN
        DO 111 R=1,NRCOMS
          FIROOM(R) = .FALSE.
          DO 112 O=1,OBJS(R)
            IF (STATE(O,R).EQ.FLAME) THEN
              ZTIG(O,R) = 0.D0
              FIROOM(R) = .TRUE.
            END IF
            TPSIO(O,R) = TAN(PI/180.D0*PSIO(O,R))
  112     CONTINUE
          LAYR(R) = FIROOM(R)
  111   CONTINUE
        CONTINUE
      END IF
C
C THE FOLLOWING CALL TO SETSYS HAS THREE EFFECTS:
C  1) FNUM, GNUM, HNUM, INUM, SIZE ARE SET TO THE VALUES APPROPRIATE
C     FOR THE SET OF EQUATIONS IN THE SYSTEM AFTER THE DISCONTINUITY.
C  2) THE ARRAYS INSYS AND PACK (STORED IN COMMON BLOCKS NOT AVAILABLE
C     TO THIS SUBROUTINE) ARE SET UP WITH THE POINTERS FOR THE NEW SET
C     OF EQUATIONS.
C  3) THE VARIABLES IN /VAR/ WHICH ARE DEFINED BUT NOT IN THE SYSTEM
C     AFTER THE DISCONTINUITY ARE GIVEN MEANINGFUL VALUES.
C
      CALL SETSYS(FNUM,GNUM,HNUM,INUM,SIZE)
C
C THE FOLLOWING IF...ELSE IF ...END IF CONSTRUCTION ATTEMPTS TO GET
C GOOD FIRST GUESSES FOR THE VALUES OF THE VARIABLES IN THE SYSTEM.
C RECALL THAT THE VARIABLES IN /VAR/ WHICH ARE DEFINED BUT NOT IN
C THE SYSTEM (AFTER THE DISCONTINUITY) NOW CONTAIN THE APPROPRIATE
C VALUES FOR AFTER THE DISCONTINUITY.  HOWEVER, A VARIABLE WHICH IS
C IN THE SYSTEM AFTER THE DISCONTINUITY CONTAINS 1) NO MEANINGFUL
C VALUE IF THE VARIABLE WAS UNDEFINED BEFORE THE DISCONTINUITY, OR
C 2) THE CONVERGED VALUE FROM BEFORE THE DISCONTINUITY IF THE VARIABLE
C WAS DEFINED BEFORE THE DISCONTINUITY.  WE WISH TO GET A COMPLETE
C SET OF CONVERGED VALUES FOR THE VARIABLES BEFORE WE RESTART THE
C INTEGRATOR.  HOWEVER, WE CANNOT SOLVE THE NEW SET OF SIMULTANEOUS
C EQUATIONS UNLESS WE HAVE A GOOD FIRST GUESS AT THE SOLUTION.  THE
C FOLLOWING IF BLOCK TRIES TO PROVIDE THAT FIRST GUESS.
C  NOTE THAT WE ONLY MAKE A FIRST GUESS FOR THE NON- ELIMINATED
C VARIABLES.  A FIRST GUESS CAN THEN BE OBTAINED FOR THE OTHER
C VARIABLES BY MAKING A SINGLE CALL TO FUNCT.
C
      IF (DSCTYP.EQ.SPLOVR) THEN
C
          ZMLZZ(RM) = 1.D-2*ZLRZX(RM)*ZLRZY(RM)*VMAZZ(RM)
          ZELZZ(RM) = ZKAZZ(RM)*CP*ZMLZZ(RM)
          ZMDL(RM) = YCO2*ZMLZZ(RM)
          ZMSL(RM) = 0.D0
          ZMML(RM) = 0.D0
          ZMOL(RM) = YO2*ZMLZZ(RM)
          ZMWL(RM) =0.D0
          TELZZ(RM) = 1.D4
      ELSE IF (DSCTYP.EQ.BRNOUT) THEN
C IF A BURNOUT HAS JUST OCCURRED, THE VALUES FOR THE VARIABLES FROM
C BEFORE THE DISCONTINUITY ARE A GOOD APPROXIMATION TO THE
C NEW CONVERGED SOLUTION.
        CONTINUE
      ELSE IF (DSCTYP.EQ.HEATIG) THEN
C IF AN IGNITION THROUGH HEATING HAS OCCURRED, THEN SET THE FIRE
```

```
C RADIUS TO THE INITIAL BURNING RADIUS AND THE FLAME HEIGHT TO
C THE HEIGHT WHICH THE NEW FIRE WOULD HAVE IF IT WERE NOT OXYGEN
C STARVED. (SINCE THE INITIAL BURNING RADIUS IS SMALL, OTHER VARIABLES
C ARE NOT SIGNIFICANTLY AFFECTED).
C
C THE FOLLOWING IF BLOCK TAKES CARE OF THE CASE WHERE THE NEW FIRE
C BEGINS A NEW HOT LAYER IN THE ROOM.
        IF (ZMLZZ(RM).EQ.0.D0) THEN
          ZMLZZ(RM) = 1.D-2*ZLRZX(RM)*ZLRZY(RM)*VMAZZ(RM)
          ZELZZ(RM) = ZKAZZ(RM)*CP*ZMLZZ(RM)
          ZMCL(RM) = YCO2*ZMLZZ(RM)
          ZMSL(RM) = 0.D0
          ZMML(RM) = 0.D0
          ZMOL(RM) = YO2*ZMLZZ(RM)
          ZMWL(RM) =0.D0
          TELZZ(RM) = 1.D4
        END IF
        ZRFZZ(OB,RM) = 1.D-3
        PHI(OB,RM) = FQLOR(OB,RM)+5.D3
        TMOZB(OB,RM) = CHI(OB,RM)*PI*ZRFZZ(OB,RM)**2*
     1         PHI(OB,RM)/CVAP(OB,RM)
      ELSE IF (DSCTYP.EQ.CNTCIG) THEN
C
C THE FOLLOWING IF BLOCK TAKES CARE OF THE CASE WHERE THE NEW FIRE
C BEGINS A NEW HOT LAYER IN THE ROOM.
        IF (ZMLZZ(RM).EQ.0.D0) THEN
          ZMLZZ(RM) = 1.D-2*ZLRZX(RM)*ZLRZY(RM)*VMAZZ(RM)
          ZELZZ(RM) = ZKAZZ(RM)*CP*ZMLZZ(RM)
          ZMCL(RM) = YCO2*ZMLZZ(RM)
          ZMSL(RM) = 0.D0
          ZMML(RM) = 0.D0
          ZMOL(RM) = YO2*ZMLZZ(RM)
          ZMWL(RM) =0.D0
          TELZZ(RM) = 1.D4
        END IF
        ZRFZZ(OB,RM) = ZRFZO(OB,RM)
        PHI(OB,RM) = FQLOR(OB,RM)+5.D3
        TMOZB(OB,RM) = CHI(OB,RM)*PI*ZRFZZ(OB,RM)**2*
     1         PHI(OB,RM)/CVAP(OB,RM)
      ELSE IF (DSCTYP.EQ.STRTUP) THEN
        DO 120 R = 1,NROOMS
          ZPRZZ(R) = 0.D0
C WE MUST START WITH A NON-ZERO LAYER MASS IN EACH ROOM BECAUSE:
C 1) THE CONCENTRATION EQUATIONS ARE SINGULAR WHEN THE LAYER MASS
C IS 0.D0, AND 2) IF WE START WITH A LAYER MASS OF 0.D0, THE LAYER
C TEMPERATURE WILL BE DISCONTINUOUS WHEN ONE ROOM STARTS TO POUR
C HOT GASSES THROUGH A VENT INTO A PREVIOUSLY UNINVOLVED ROOM.
C WE THEREFORE START WITH A LAYER DEPTH OF 1 CM IN EACH ROOM.
          IF (LAYR(R)) THEN
            ZMLZZ(R) = 1.D-2*ZLRZX(R)*ZLRZY(R)*VMAZZ(R)
            ZMCL(R) = ZMLZZ(R)*YCO2
            ZMML(R) = 0.D0
            ZMOL(R) = ZMLZZ(R)*YO2
            ZMSL(R) = 0.D0
            ZMWL(R) = 0.D0
            ZELZZ(R) = CP*ZKAZZ(R)*ZMLZZ(R)
            TELZZ(R) = 1.D4
          END IF
C    VARIABLES INDEXED BY OBJECT WITHIN THE ROOM:
          DO 110 O = 1,OBJS(R)
            IF (STATE(O,R).EQ.FLAME) ZMOZZ(O,R) = ZMOZO(O,R)
C IF THE OBJECT IS NOT BURNING, THEN ZMOZZ HAS BEEN INITIALIZED
C IN SETSYS.
```

```
          IF (STATE(O,R) .EQ. HEATNG) ZKOZZ(O,R) = ZKAZZ(R)
          IF (STATE(O,R) .EQ. FLAME) THEN
           IF (FIRTYP(O,R).EQ.GROWNG) THEN
            ZRFZZ(O,R) = ZRFZO(O,R)
           ELSE
            ZRFZZ(O,R) = 1.D-3
           END IF
            PHI(O,R) = 5.D3
          TMOZB(O,R) = CHI(O,R)*PI*ZRFZZ(O,R)**2*PHI(O,R)/QVAP(O,R)
          END IF
  110     CONTINUE
  120   CONTINUE
C     VARIABLES INDEXED BY WALL:
        DO 140 W = 1,NWALLS
          AQWZZ(W) = 0.D0
          DO 130 SIDE = 1,2
            IF (WLINK(SIDE,W).EQ.0) THEN
              ZKA = ZKAOUT
            ELSE
              ZKA = ZKAZZ(WLINK(SIDE,W))
            END IF
            ZKWZZ(SIDE,W) = ZKA
  130     CONTINUE
  140   CONTINUE
      END IF
C
C NOW WE USE THE NUMERICS TO TRY TO GET A SET OF CONVERGED VALUES
C FOR THE VARIABLES.
C  FIRST, PACK THE ESTIMATED VALUES FOR THE VARIABLES INTO THE X
C  VECTOR:
      CALL PACKX(X)
C
C NOW CALL THE PHYSICAL SUBROUTINES TO GET ESTIMATED VALUES FOR
C THE ELIMINATED VARIABLES.
      CALL FUNCT(X,ZERO,TIME)
C
C COPY THE OUTPUT OF THE ELIMINATED VARIABLE EQUATIONS INTO THE
C PROPER PLACES IN X.
      DO 150 C=FNUM+GNUM+HNUM+1,SIZE
        X(C) = ZERO(C)
  150 CONTINUE
C
C X NOW CONTAINS A COMPLETE SET OF ESTIMATES FOR THE PHYSICAL
C VARIABLES.  WE NOW ATTEMPT TO GET AN EXACT SOLUTION.
C
C INFORM THE USER THAT PROGRAM IS ABOUT TO TRY TO GET
C THROUGH THE DISCONTINUITY.
      IF (DSCTYP.EQ.BRNOUT) THEN
        WRITE(IWDSK,151) OB,RM,TIME
        WRITE(IWTTY,151) OB,RM,TIME
  151   FORMAT(X,'ATTEMPTING TO BURN OUT OBJECT,'I2,' IN ROOM',I2,
     1   ' AT TIME',F7.2,'.')
      ELSE IF(DSCTYP.EQ.HEATIG) THEN
        WRITE(IWDSK,152) OB,RM,TIME
        WRITE(IWTTY,152) OB,RM,TIME
  152   FORMAT(X,'ATTEMPTING TO IGNITE OBJECT',I2,' IN ROOM',I2,
     1   ' AT TIME',F7.2,' DUE TO HEATING.')
      ELSE IF(DSCTYP.EQ.CNTCIG) THEN
        WRITE(IWDSK,153) OB,RM,TIME
        WRITE(IWTTY,153) OB,RM,TIME
  153   FORMAT(X,'ATTEMPTING TO IGNITE OBJECT',I2,' IN ROOM',I2,
     1 ' AT TIME',F7.2,' DUE TO CONTACT WITH FLAMES.')
      ELSE IF(DSCTYP.EQ.STRTUP) THEN
```

```
          WRITE(IWDSK,154)
          WRITE(IWTTY,154)
  154     FORMAT(X,'ATTEMPTING TO GET INITIAL VALUES FOR THE RUN.')
        ELSE IF(DSCTYP.EQ.SPLOVR) THEN
          WRITE(IWDSK,1546) RM
          WRITE(IWTTY,1546) RM
 1546     FORMAT(' ATTEMPTING TO INITIALIZE THE LAYER IN ROOM ',I2)
        END IF
C
        CALL SCALE(X,XMAX)
C
C SCALE THE FIRST TIME DERIVATIVES IN X FOR A SIZE 0.D0 TIME STEP.
        DO 1545 O=SIZE+1,2*SIZE
          X(O) = 0.D0
 1545   CONTINUE
C
        CALL EVJAC(FNUM,X,EPS,XMAX,TIME,FNUM,FNUM+1,FNUM+GNUM,0.D0,
     1     FNUM+GNUM+1,1.D0,SIZE,MAXVR1,PIVOT,SING,JAC,Z,ZERO)
        IF (SING) THEN
C THE GAUSS-SEIDEL MATRIX WAS SINGULAR.
          WRITE(IWDSK,155)
          WRITE(IWTTY,155)
  155     FORMAT(X,'THE GAUSS SEIDEL MATRIX IS SINGULAR.')
        ELSE
C MATRIX O.K.  TRY TO SOLVE EQS.
         CALL CORECT(JAC,200,TIME,X,FNUM+1,FNUM+GNUM,SIZE,0.D0,
     2   FNUM+GNUM+1,MAXVR1,FNUM,PIVOT,FNUM+1,1.D0,Z,FNUM,
     3   EPS/(6.D0*DBLE(SIZE)),XMAX,CONV,0,ZERO)
         IF (CONV) THEN
           RETURN
         ELSE
           WRITE(IWTTY,190)
           WRITE(IWDSK,190)
  190      FORMAT(X,'GAUSS SEIDEL COULD NOT SOLVE THE EQUATIONS.')
         END IF
        END IF
C COME HERE IF THE GAUSS-SEIDEL METHOD DOESN'T WORK.
        CALL EVJAC(FNUM+GNUM+HNUM,X,EPS,XMAX,TIME,FNUM,FNUM+1,
     1     FNUM+GNUM,0.D0,
     1     FNUM+GNUM+1,1.D0,SIZE,MAXVR1,PIVOT,SING,JAC,Z,ZERO)
        IF (SING) THEN
C THE NEWTON MATRIX IS SINGULAR.
          WRITE(IWDSK,157)
          WRITE(IWTTY,157)
  157     FORMAT(X,'THE NEWTON MATRIX IS SINGULAR.  BOMBING OUT.')
        ELSE
C THE MATRIX IS O.K.  TRY TO SOLVE THE EQUATIONS.
         CALL CORECT(JAC,10,TIME,X,FNUM+1,FNUM+GNUM,SIZE,0.D0,
     2   FNUM+GNUM+1,MAXVR1,FNUM+GNUM+HNUM,PIVOT,FNUM+1,1.D0,Z,FNUM,
     3   EPS/(6.D0*DBLE(SIZE)),XMAX,CONV,0,ZERO)
         IF (CONV) THEN
           RETURN
         ELSE
           WRITE(IWTTY,160)
           WRITE(IWDSK,160)
  160      FORMAT(X,'NEWTON''S METHOD COULD NOT SOLVE THE EQUATIONS.
     1    BOMBING OUT.')
         END IF
        END IF
        STOP
        END
        SUBROUTINE SCALE(X,XMAX)
        IMPLICIT INTEGER (A-Z)
```

```
      REAL*8 XMAX(*),X(*)
C
C+ PART OF MODULE INTERFACE.
C RETURNS A VALUE FOR EACH PHYSICAL VARIABLE THAT REPRESENTS
C THE MAGNITUDE OF A SIGNIFICANT VALUE OF THAT VARIABLE.
C
      SAVE
$INCLUDE 9,DIMN.CMN
      INTEGER INSYS
      COMMON /SYSPTR/ INSYS(MAXVRS+1)
      INTEGER INDEX
      REAL*8 XSCALE
C
C COMMON /SCALER/ CONTAINS CONSTANTS THAT REPRESENT THE SMALLEST
C VALUE OF SIGNIFICANCE OF VARIABLES IN COMMON /VAR/.
      COMMON /SCALER/ XSCALE(NVARS)
C
C COMMON /SCALER/ IS ACCESSED USING XSCALE AND INSYS
C FOR ALL VARIABLES.
      INDEX = 1
C
C WHILE INSYS(INDEX)<>0 DO
   10 IF (INSYS(INDEX) .EQ. 0) GO TO 20
        XMAX(INDEX) = MAX(ABS(X(INDEX)),XSCALE(INSYS(INDEX)))
        INDEX = INDEX + 1
        GO TO 10
   20 CONTINUE
      RETURN
      END
      SUBROUTINE SETSYS(FNUM,GNUM,HNUM,INUM,SIZE)
      IMPLICIT INTEGER (A-Z)
      INTEGER FNUM,GNUM,HNUM,INUM,SIZE
C
C+       PART OF MODULE INTERFACE
C
C THIS SUBROUTINE SHOULD BE CALLED AT THE BEGINNING OF EACH TIME
C STEP TO DETERMINE WHICH VARIABLES ARE IN THE SYSTEM, THAT IS
C WHICH VARIABLES ARE PHYSICALLY MEANINGFUL AND POTENTIALLY
C VARYING AT THE CURRENT POINT IN THE CALCULATION.  THE SUBROU-
C TINE LOOKS AT VARIOUS FLAGS TO DETERMINE WHICH VARIABLES ARE
C IN THE SYSTEM THEN SETS UP THE POINTER ARRAYS INSYS AND PACK.
C INSYS IS AN ARRAY OF POINTERS INTO THE FIRST SECTION OF THE
C COMMON BLOCK /VAR/ WHICH POINTS TO THE VARIABLES THAT ARE
C CURRENTLY IN THE SYSTEM.  THE ORDER OF THE POINTERS IN INSYS
C DETERMINES THE ORDER IN WHICH THE VARIABLES ARE PACKED FROM
C /VAR/ INTO THE NUMERICAL VECTOR X IN ALDIFF.  PACK IS AN
C ARRAY OF POINTERS INTO /VAR/ WHICH POINTS TO THOSE VARIABLES
C WHICH ARE THE OUTPUT OF THE EQUATIONS NEEDED TO SOLVE FOR THE
C VARIABLES IN THE SYSTEM.  THUS, FUNCT USES THE ARRAY PACK TO
C SELECT THE VARIABLES AND THE ORDER FOR REPACKING INTO THE
C ARRAY ZERO.
C
C RECALL THAT THE NUMERICAL PROBLEM CONSISTS OF ROOT FINDING
C PROBLEMS, DIFFERENTIAL EQUATIONS, FIXED POINT EQUATIONS
C AND ELIMINATED VARIABLE EQUATIONS, I.E. THE PROBLEM HAS THE
C FORM
C                   0 = F(U,V,W,T)
C                   V'= G(U,V,W,T)
C                   W = H(U,V,W,T)
C                   Y = I(U,V,W,T)
C THEN INSYS STORES THE POINTERS TO U,V,W,Y CONTIGUOUSLY AND
C IN THAT ORDER.  PACK STORES THE POINTERS TO THE OUTPUT OF
C THE VECTOR VALLED FUNCTIONS F,G,H,I CONTIGUOUSLY AND IN THAT
```

```
C ORDER.
C
C THE FOLLOWING LINES SET UP A BLOCK OF PARAMETERS OF POINTERS
C INTO /VAR/.  FOR EXAMPLE, PTELZZ+1 POINTS TO TELZZ(1) IN /VAR/.
C SINCE MANY OF THE VARIABLE NAMES IN SECTION 3 OF VAR ALREADY
C HAVE 6 CHARACTERS, THE POINTER NAME FOR THESE VARIABLES DROPS
C THE D PREFIX AND REPLACES IT WITH A Q PREFIX.
C
$INCLUDE 9,DIMN.CMN
$INCLUDE 9,BLDNG.CMN
$INCLUDE 9,VAR6.CMN
      INTEGER INSYS,PACK,VARY,FIXED,REGIM1,REGIM2,REGIM3
      PARAMETER (FIXED = 0, REGIM1 = 1, REGIM2 = 2, REGIM3 = 3)
      COMMON /VARYNG/ VARY(NVARS)
      COMMON /SYSPTR/ INSYS(MAXVRS+1)
      COMMON /PCKPTR/ PACK(MAXVRS+1)

C
      INTEGER PTELZR,PTELZD,PZMLZZ,PTMLZZ,PZELZZ,
     1 PTELZZ,PZHLZZ,PZKLZZ,PZYLCZ,PZYLDZ,PZYLMZ,
     2 PZYLSZ,PZYLWZ,PZPRZZ,PZULZZ,PFQLWR,PFQPWR,
     3 PFQLWD,PTMODS,PTMOMS,PTMOSS,PTMOWS,PZMDL,
     4 PZMML,PZMOL,PZMWL,PZMSL,PAVLSM,PWVLS,
     5 PPPLR,PTMCSM,PTMPSM,PTMUSM,PTMDSM,PTELSM,
     6 PEPLSM,PBCXYS,PTMDZF,PTMWZF,PTMSZF,PTMMZF,
     + PTMOZF,PFQPOR,PZKOZZ,PZMOZZ,
     7 PTMOZZ,PTEOZZ,PZHPZZ,PTMPZZ,
     8 PTEPZR,PZRFZZ,PZHFZZ,PFQLOR,PTEPSM,PTEOSM,
     9 PFQWOR,PTMOZB,PPHI,PAQWZZ,PZKWZZ,
     1 PTMNET,
     2 QZMLZZ,QZELZZ,QZMOZZ,QZMDL,QZMML,
     3 QZMOL,QZMWL,QZMSL,QAQWZZ,QZRFZZ
      INTEGER DUM,O,R,W,V
C
C
      PARAMETER (PTELZR=0,
     1           PTELZD=PTELZR+MAXRMS,
     2           PZMLZZ=PTELZD+MAXRMS,
     3           PTMLZZ=PZMLZZ+MAXRMS,
     4           PZELZZ=PTMLZZ+MAXRMS,
     5           PTELZZ=PZELZZ+MAXRMS,
     6           PZHLZZ=PTELZZ+MAXRMS,
     7           PZKLZZ=PZHLZZ+MAXRMS,
     8           PZYLOZ=PZKLZZ+MAXRMS,
     9           PZYLDZ=PZYLOZ+MAXRMS)
      PARAMETER (PZYLMZ=PZYLDZ+MAXRMS,
     1           PZYLSZ=PZYLMZ+MAXRMS,
     2           PZYLWZ=PZYLSZ+MAXRMS,
     3           PZPRZZ=PZYLWZ+MAXRMS,
     4           PZULZZ=PZPRZZ+MAXRMS,
     5           PTEPSM=PZULZZ+MAXRMS,
     6           PTEOSM=PTEPSM+MAXRMS,
     7           PTMODS=PTEOSM+MAXRMS,
     8           PTMOMS=PTMODS+MAXRMS)
      PARAMETER (PTMOSS=PTMOMS+MAXRMS,
     1           PTMOWS=PTMOSS+MAXRMS,
     2           PZMDL =PTMOWS+MAXRMS,
     3           PZMML =PZMDL+MAXRMS,
     4           PZMOL =PZMML+MAXRMS,
     5           PZMWL =PZMOL+MAXRMS,
     6           PZMSL =PZMWL+MAXRMS,
     7           PAVLSM=PZMSL+MAXRMS,
     8           PWVLS =PAVLSM+MAXRMS,
     9           PPPLR  =PWVLS+MAXRMS)
      PARAMETER (PTMOSM=PPPLR+MAXRMS,
     1           PTMPSM=PTMOSM+MAXRMS,
     2           PTMUSM=PTMPSM+MAXRMS,
     3           PTMDSM=PTMUSM+MAXRMS,
     4           PTEUSM=PTMDSM+MAXRMS,
     5           PEPLSM=PTEUSM+MAXRMS,
     6           PBOXYS=PEPLSM+MAXRMS,
     7           PTMDZF=PBOXYS+MAXRMS,
     8           PTMWZF=PTMDZF+MAXRMS,
     9           PTMSZF=PTMWZF+MAXRMS,
     +           PTMMZF=PTMSZF+MAXRMS,
     1           PTMOZF=PTMMZF+MAXRMS)
**********************************************************************
      PARAMETER (PFQPOR=PTMOZF+MAXRMS,
     1           PZKOZZ=PFQPOR+MAXRMS*MAXOBS,
     2           PZMOZZ=PZKOZZ+MAXRMS*MAXOBS,
     3           PTMOZZ=PZMOZZ+MAXRMS*MAXOBS,
     4           PTEOZZ=PTMOZZ+MAXRMS*MAXOBS,
     5           PZHPZZ=PTEOZZ+MAXRMS*MAXOBS,
     6           PTMPZZ=PZHPZZ+MAXRMS*MAXOBS,
     7           PTEPZR=PTMPZZ+MAXRMS*MAXOBS,
     8           PZRFZZ=PTEPZR+MAXRMS*MAXOBS,
     9           PZHFZZ=PZRFZZ+MAXRMS*MAXOBS)
      PARAMETER (PFQLOR=PZHFZZ+MAXRMS*MAXOBS,
     1           PFQWOR=PFQLOR+MAXRMS*MAXOBS,
     2           PTMOZB=PFQWOR+MAXRMS*MAXOBS,
     3           PPHI  =PTMOZB+MAXRMS*MAXOBS)
****************************************************************
      PARAMETER (PAQWZZ=PPHI  +MAXRMS*MAXOBS,
     1           PZKWZZ=PAQWZZ+MAXWLS,
     2           PFQLWR = PZKWZZ + 2*MAXWLS,
     3           PFQPWR = PFQLWR + 2*MAXWLS,
     4           PFQLWD = PFQPWR + 2*MAXWLS)
****************************************************************
      PARAMETER (PTMNET=PFQLWD+2*MAXWLS)
****************************************************************
      PARAMETER (QZMLZZ=PTMNET+MAXRMS,
     1           QZELZZ=QZMLZZ+MAXRMS,
     2           QZMOZZ=QZELZZ+MAXRMS,
     3           QZMDL =QZMOZZ+MAXRMS*MAXOBS,
     4           QZMML =QZMDL+MAXRMS,
     5           QZMOL =QZMML+MAXRMS,
     6           QZMWL =QZMOL+MAXRMS,
     7           QZMSL =QZMWL+MAXRMS,
     8           QAQWZZ=QZMSL+MAXRMS,
     9           QZRFZZ=QAQWZZ+MAXWLS,
     +           DUM =  QZRFZZ+MAXRMS*MAXOBS)
C
C
C
      FNUM = 0
      GNUM = 0
      HNUM = 0
      INUM = 0
      SIZE = 0
      INDEX = 0
      DO 1 R=1,NVARS
        VARY(R) = FIXED
    1 CONTINUE
C
C DECIDE WHICH VARIABLES ARE VARYING ACCORDING TO WHICH REGIMES AND
C WHICH ARE FIXED.  SET THE VARIABLES THAT ARE FIXED TO THE APPROPRIATE
```

```
C VALUES.
C
C ZPRZZ
      DO 10 R=1,NROOMS
        VARY(PZPRZZ+R) = REGIM1
   10 CONTINUE
C
C ZMLZZ
      DO 70 R=1,NROOMS
        IF (LAYR(R)) THEN
          VARY(PZMLZZ+R) = REGIM1
C
C ZELZZ
          VARY(PZELZZ+R) = REGIM1
C
C ZMDL,ZMML,ZMOL,ZMWL,ZMSL
          VARY(PZMDL+R) = REGIM1
          VARY(PZMML+R) = REGIM1
          VARY(PZMOL+R) = REGIM1
          VARY(PZMWL+R) = REGIM1
          VARY(PZMSL+R) = REGIM1
        ELSE
          ZMLZZ(R) = 0.D0
          ZELZZ(R) = 0.D0
          ZMDL(R) = 0.D0
          ZMML(R) = 0.D0
          ZMOL(R) = 0.D0
          ZMSL(R) = 0.D0
          ZMWL(R) = 0.D0
        END IF
C
C ZMOZZ
        DO 50 O=1,OBJS(R)
          IF (STATE(O,R).EQ.FLAME) THEN
            VARY(PZMOZZ+(R-1)*MAXOBS+O) = REGIM1
          ELSE IF (STATE(O,R).EQ.COLD .OR. STATE(O,R).EQ.HEATNG) THEN
            ZMOZZ(O,R) = ZMOZO(O,R)
          ELSE IF (STATE(O,R).EQ.CNSUM) THEN
            ZMOZZ(O,R) = 0.D0
          END IF
C
C ZRFZZ
          IF (STATE(O,R) .EQ. FLAME) THEN
            IF (FIRTYP(O,R) .EQ. GROWNG) THEN
              VARY(PZRFZZ + (R-1)*MAXOBS + O) = REGIM1
            ELSE IF (FIRTYP(O,R) .EQ. POOL) THEN
              VARY(PZRFZZ + (R-1)*MAXOBS + O) = REGIM2
            END IF
           ELSE
            ZRFZZ(O,R) = 0.D0
           END IF
   50   CONTINUE
C
   70 CONTINUE
C
C AQWZZ
      DO 80 W=1,NWALLS
        VARY(PAQWZZ+W) = REGIM1
   80 CONTINUE
C
C ZKLZZ,TELZC
C
      DO 100 R=1,NROOMS
```

```
        IF (LAYR(R)) THEN
          VARY(PZKLZZ+R) = REGIM1
          VARY(PTELZD+R) = REGIM1
        ELSE
          ZKLZZ(R) = ZKAZZ(R)
          TELZD(R) = 0.D0
        END IF
C
        DO 90 O=1,OBJS(R)
C
C FQPOR
          IF (STATE(O,R).NE.CNSUM) THEN
            VARY(PFQPOR+(R-1)*MAXOBS+O) = REGIM1
          ELSE
            FQPOR(O,R) = 0.D0
          END IF
C
C
C ZKOZZ
          IF (STATE(O,R).EQ.HEATNG) THEN
            IF (LAYR(R)) THEN
              VARY(PZKOZZ +  (R-1)*MAXOBS + O) = REGIM1
            ELSE
              ZKOZZ(O,R) = ZKAZZ(R)
            END IF
          ELSE IF (STATE(O,R) .EQ. FLAME) THEN
            ZKOZZ(O,R) = ZKOIG(O,R)
          ELSE IF (STATE(O,R).EQ.COLD .OR. STATE(O,R).EQ.CNSUM) THEN
            ZKOZZ(O,R) = ZKAZZ(R)
          END IF
C
C TMOZB,ZHFZZ
          IF (STATE(O,R).EQ.FLAME) THEN
C
            VARY(PTMOZB +  (R-1)*MAXOBS + O) = REGIM1
            VARY(PZHFZZ + (R-1)*MAXOBS + O) = REGIM1
C
          ELSE
            TMOZB(O,R) = 0.D0
            ZHFZZ(O,R) = 0.D0
          END IF
   90   CONTINUE
  100 CONTINUE
C
C ZKWZZ
      DO 120 W=1,NWALLS
        DO 110 SIDE=1,2
          PTR = PZKWZZ + 2*(W-1) + SIDE
          VARY(PTR) = REGIM1
  110   CONTINUE
  120 CONTINUE
C
C TELZR,TMLZZ,TELZZ,ZHLZZ,ZYLCZ,ZYLDZ,ZYLMZ,ZYLSZ,
C ZYLWZ,ZULZZ,AVLSM,WVLS,TMUSM,TMDSM,TEUSM
      DO 140 R=1,NROOMS
C
        IF (LAYR(R)) THEN
          PTR = PTELZR+R
          VARY(PTR) = REGIM1
          PTR = PTMLZZ+R
          VARY(PTR) = REGIM1
          PTR = PTELZZ+R
          VARY(PTR) = REGIM1
          PTR = PZHLZZ+R
          VARY(PTR) = REGIM1
          PTR = PZYLCZ+R
          VARY(PTR) = REGIM1
          PTR = PZYLDZ+R
          VARY(PTR) = REGIM1
          PTR = PZYLMZ+R
          VARY(PTR) = REGIM1
          PTR = PZYLSZ+R
          VARY(PTR) = REGIM1
          PTR = PZYLWZ+R
          VARY(PTR) = REGIM1
          PTR = PZULZZ+R
          VARY(PTR) = REGIM1
          PTR = PAVLSM+R
          VARY(PTR) = REGIM1
        ELSE
          TELZR(R) = 0.D0
          TMLZZ(R) = 0.D0
          TELZZ(R) = 0.D0
          ZHLZZ(R) = 0.D0
          ZYLCZ(R) = 0.D0
          ZYLDZ(R) = 0.D0
          ZYLMZ(R) = 0.D0
          ZYLSZ(R) = 0.D0
          ZYLWZ(R) = 0.D0
          ZULZZ(R) = 0.D0
          AVLSM(R) = 0.D0
        END IF
        PTR = PWVLS+R
        VARY(PTR) = REGIM1
        PTR = PTMUSM+R
        VARY(PTR) = REGIM1
        PTR = PTMCSM+R
        VARY(PTR) = REGIM1
        PTR = PTEUSM+R
        VARY(PTR) = REGIM1
C
C TMOCS,TMOMS,TMOSS,TMOWS,PPLR,TMOSM,TMPSM,EPLSM,BOXYS,TEPSM,TEOSM
        IF (FIROCM(R)) THEN
          PTR = PTMOCS+R
          VARY(PTR) = REGIM1
          PTR = PTMOMS + R
          VARY(PTR) = REGIM1
          PTR = PTMOSS+R
          VARY(PTR) = REGIM1
          PTR = PTMOWS+R
          VARY(PTR) = REGIM1
          PTR = PPPLR+R
          VARY(PTR) = REGIM1
          PTR = PTMOSM+R
          VARY(PTR) = REGIM1
          PTR = PTMPSM+R
          VARY(PTR) = REGIM1
          PTR = PEPLSM+R
          VARY(PTR) = REGIM1
          PTR = PBOXYS+R
          VARY(PTR) = REGIM1
          PTR = PTEPSM + R
          VARY(PTR) = REGIM1
          PTR = PTEOSM + R
          VARY(PTR) = REGIM1
        ELSE
```

```
     TMCDS(R) = 0.D0
     TMCMS(R) = 0.D0
     TMCWS(R) = 0.D0
     PPLR(R) = 0.D0
     TMCSM(R) = 0.D0
     TMFSM(R) = 0.D0
     EPLSM(R) = 0.D0
     BOXYS(R) = 0.DC
     TEFSM(R) = 0.DC
     TECSM(R) = 0.D0
    END IF
C
    DO 130 O=1,OBJS(R)
C
C FQLOR,FQWOR
     IF (STATE(O,R).NE.CNSUM) THEN
      IF (LAYR(R)) THEN
       PTR = PFQLOR +  (R-1)*MAXOBS + O
       VARY(PTR) = REGIM1
      ELSE
       FQLOR(O,R) = 0.D0
      END IF
      PTR = PFQWOR +  (R-1)*MAXOBS + O
      VARY(PTR) = REGIM1
     ELSE
      FQLOR(O,R) = 0.D0
      FQWOR(O,R) = 0.DC
     END IF
C
C
C TMOZZ,TEOZZ,TMPZZ,TEPZR,PHI
     IF (STATE(O,R).EQ.FLAME) THEN
      PTR = PTMOZZ +  (R-1)*MAXOBS + O
      VARY(PTR) = REGIM1
      PTR = PTEOZZ +  (R-1)*MAXOBS + O
      VARY(PTR) = REGIM1
      PTR = PTMPZZ +  (R-1)*MAXOBS + O
      VARY(PTR) = REGIM1
      PTR = PTEPZR +  (R-1)*MAXOBS + O
      VARY(PTR) = REGIM1
      PTR = PPHI +  (R-1)*MAXOBS + O
      VARY(PTR) = REGIM1
     ELSE
      TMOZZ(O,R) = 0.D0
      TEOZZ(O,R) = 0.DC
      TMPZZ(O,R) = 0.D0
      TEPZR(O,R) = 0.D0
      PHI(O,R) = 0.D0
     END IF
C
C ZHPZZ
     IF (LAYR(R)) THEN
      PTR = PZHPZZ +  (R-1)*MAXOBS + O
      VARY(PTR) = REGIM1
     ELSE
      ZHPZZ(O,R) = ZHRZZ(R) - ZHOZZ(O,R)
     END IF
 130    CONTINUE
 140  CONTINUE
C
C
C FQLWD, FQLWR, FQPWR
   DO 160 W=1,NWALLS
```

```
        DO 150 SIDE=1,2
          PTR = PFQLWC + 2*(W-1)+SIDE
          VARY(PTR) = REGIM1
          R = WLINK(SIDE,W)
          IF (R .NE. 0 .AND. LAYR(R)) THEN
            PTR = PFQLWR + 2*(W-1) + SIDE
            VARY(PTR) = REGIM1
          ELSE
            FQLWR(SIDE,W) = 0.D0
          END IF
C
          R = WLINK(SIDE,W)
C
C NOTE THAT IN THE FOLLOWING CONDITIONAL, SWITCHING THE ORDER OF THE
C R.NE.0 WITH THE FIROOM(R) WILL CAUSE A MEMORY FAULT IN THE CASE
C WHERE R IS 0.
          IF (R.NE.0 .AND. FIROOM(R)) THEN
            PTR = PFQPWR + 2*(W-1) + SIDE
            VARY(PTR) = REGIM1
          ELSE
            FQPWR(SIDE,W) = 0.D0
          END IF
  150   CONTINUE
  160 CONTINUE
C
C NOW SET UP THE ARRAYS PACK AND INSYS:
C  ROOT FINDING EQUATIONS AND VARIABLES:
      DO 170 R=1,NROOMS
        INDEX = R
        CALL PUTPTR(PZPRZZ,PTMNET,INDEX,REGIM1,SIZE,FNUM)
  170 CONTINUE
C
C  DIFFERENTIAL EQUATIONS AND VARIABLES:
      DO 180 R=1,NROOMS
        INDEX = R
        CALL PUTPTR(PZELZZ,QZELZZ,INDEX,REGIM1,SIZE,GNUM)
        CALL PUTPTR(PZMLZZ,QZMLZZ,INDEX,REGIM1,SIZE,GNUM)
        CALL PUTPTR(PZMCL,QZMCL,INDEX,REGIM1,SIZE,GNUM)
        CALL PUTPTR(PZMSL,QZMSL,INDEX,REGIM1,SIZE,GNUM)
        CALL PUTPTR(PZMML,QZMML,INDEX,REGIM1,SIZE,GNUM)
        CALL PUTPTR(PZMOL,QZMOL,INDEX,REGIM1,SIZE,GNUM)
        CALL PUTPTR(PZMWL,QZMWL,INDEX,REGIM1,SIZE,GNUM)
        DO 175 O=1,OBJS(R)
          INDEX = (R-1)*MAXOBS+O
          CALL PUTPTR(PZRFZZ,QZRFZZ,INDEX,REGIM1,SIZE,GNUM)
          CALL PUTPTR(PZMCZZ,QZMCZZ,INDEX,REGIM1,SIZE,GNUM)
  175   CONTINUE
  180 CONTINUE
      DO 190 W=1,NWALLS
        INDEX=W
        CALL PUTPTR(PAQWZZ,QAQWZZ,INDEX,REGIM1,SIZE,GNUM)
  190 CONTINUE
C
C FIXED POINT VARIABLES (NON-ELIMINATED):
      DO 200 R=1,NROOMS
        INDEX = R
        CALL PUTPTR(PTELZZ,PTELZZ,INDEX,REGIM1,SIZE,HNUM)
        DO 195 O=1,OBJS(R)
          INDEX = (R-1)*MAXOBS + O
          CALL PUTPTR(PPHI,PPHI,INDEX,REGIM1,SIZE,HNUM)
          CALL PUTPTR(PTMCZB,PTMCZB,INDEX,REGIM1,SIZE,HNUM)
          CALL PUTPTR(PZRFZZ,PZRFZZ,INDEX,REGIM2,SIZE,HNUM)
          CALL PUTPTR(PZKCZZ,PZKCZZ,INDEX,REGIM1,SIZE,HNUM)
  195   CONTINUE
  200 CONTINUE
      DO 210 W=1,NWALLS
        DO 205 SIDE=1,2
          INDEX = 2*(W-1)+SIDE
          CALL PUTPTR(PZKWZZ,PZKWZZ,INDEX,REGIM1,SIZE,HNUM)
  205   CONTINUE
  210 CONTINUE
C
C ELIMINATED VARIABLES AND EQUATIONS (ONLY THE VARIABLES WHICH MUST
C BE CHECKED FOR CONVERGENCE ARE SENT TO THE NUMERICS).
      DO 220 R=1,NROOMS
        INDEX = R
        CALL PUTPTR(PAVLSM,PAVLSM,INDEX,REGIM1,SIZE,INUM)
        CALL PUTPTR(PWVLS,PWVLS,INDEX,REGIM1,SIZE,INUM)
        CALL PUTPTR(PTELZC,PTELZC,INDEX,REGIM1,SIZE,INUM)
        CALL PUTPTR(PPPLR,PPPLR,INDEX,REGIM1,SIZE,INUM)
        CALL PUTPTR(PTELZR,PTELZR,INDEX,REGIM1,SIZE,INUM)
        CALL PUTPTR(PTEPSM,PTEPSM,INDEX,REGIM1,SIZE,INUM)
        CALL PUTPTR(PTMUSM,PTMUSM,INDEX,REGIM1,SIZE,INUM)
        CALL PUTPTR(PTMDSM,PTMDSM,INDEX,REGIM1,SIZE,INUM)
        CALL PUTPTR(PTEUSM,PTEUSM,INDEX,REGIM1,SIZE,INUM)
        CALL PUTPTR(PTMPSM,PTMPSM,INDEX,REGIM1,SIZE,INUM)
        CALL PUTPTR(PTMLZZ,PTMLZZ,INDEX,REGIM1,SIZE,INUM)
        CALL PUTPTR(PTMODS,PTMODS,INDEX,REGIM1,SIZE,INUM)
        CALL PUTPTR(PTMOSM,PTMOSM,INDEX,REGIM1,SIZE,INUM)
        CALL PUTPTR(PBOXYS,PBOXYS,INDEX,REGIM1,SIZE,INUM)
        CALL PUTPTR(PTEOSM,PTEOSM,INDEX,REGIM1,SIZE,INUM)
C
        DO 215 O=1,OBJS(R)
          INDEX = (R-1)*MAXOBS + O
          CALL PUTPTR(PTMOZZ,PTMOZZ,INDEX,REGIM1,SIZE,INUM)
          CALL PUTPTR(PZHFZZ,PZHFZZ,INDEX,REGIM1,SIZE,INUM)
          CALL PUTPTR(PFQLOR,PFQLOR,INDEX,REGIM1,SIZE,INUM)
          CALL PUTPTR(PZHPZZ,PZHPZZ,INDEX,REGIM1,SIZE,INUM)
          CALL PUTPTR(PFQPOR,PFQPOR,INDEX,REGIM1,SIZE,INUM)
          CALL PUTPTR(PFQWOR,PFQWOR,INDEX,REGIM1,SIZE,INUM)
          CALL PUTPTR(PTEFZR,PTEFZR,INDEX,REGIM1,SIZE,INUM)
  215   CONTINUE
  220 CONTINUE
      DO 230 W=1,NWALLS
        DO 225 SIDE=1,2
          INDEX = (W-1)*2 + SIDE
          CALL PUTPTR(PFQLWD,PFQLWD,INDEX,REGIM1,SIZE,INUM)
          CALL PUTPTR(PFQPWR,PFQPWR,INDEX,REGIM1,SIZE,INUM)
          CALL PUTPTR(PFQLWR,PFQLWR,INDEX,REGIM1,SIZE,INUM)
  225   CONTINUE
  230 CONTINUE
      PACK(SIZE+1) = 0
      INSYS(SIZE+1) = 0
      RETURN
      END
      SUBROUTINE PUTPTR(VPTR,FPTR,OFFSET,REGIM,SIZE,DIMN)
      IMPLICIT INTEGER (A-Z)
      INTEGER VPTR,FPTR,OFFSET,REGIM,SIZE,DIMN
C
C THIS SUBROUTINE IS CALLED BY SETSYS TO HELP SET UP THE ARRAYS PACK
C AND INSYS.
C
$INCLUDE 9,DIMN.CMN
      INTEGER PACK,INSYS,VARY
      COMMON /PCKPTR/ PACK(MAXVRS+1)
      COMMON /SYSPTR/ INSYS(MAXVRS+1)
      COMMON /VARYNG/ VARY(NVARS)
C
      IF (VARY(VPTR+OFFSET).EQ.REGIM) THEN
        SIZE = SIZE + 1
        INSYS(SIZE) = VPTR+OFFSET
        PACK(SIZE) = PPTR+OFFSET
        DIMN = DIMN + 1
      END IF
      RETURN
      END
C
C NOTE THAT IF THE PHYSICAL SUBROUTINES ARE COMPILED WITH D<LINES
C THEN LIMIT CHECKING IS DONE.  OTHERWISE, NO LIMIT CHECKING.
CHECKED
      SUBROUTINE ABSRB1(ZTZZZ,ZULZZ1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C THIS SUBROUTINE CALCULATES THE INFRA-RED ABSORPTIVITY OF THE HOT
C LAYER.  IN THIS VERSION (USED IN CFC III), WE APPROXIMATED IT BY A
C SIMPLE EXPONENTIAL FIT TO THE DATA OF THE SMOKE ATTENUATION COEFFICIEN
C OBTAINED FROM THE THIRD FULL-SCALE BEDROOM TEST (1975), CHANNEL 153.
C IN CFC III, HOWEVER, WE USED ITS RECIPROCAL, THE ABSORPTION LENGTH
C LAMDA.
C
C EQUATIONS BY H. MITLER, JULY 1979
C CODED BY B. LONDON, DECEMBER 1979
C ADAPTED TO NEW PHYSICAL SUBROUTINE FORMAT BY J. GAHM, JULY 1981.
C
C INPUT PARAMETERS:
C       ZTZZZ -- TIME.
C
C INPUT VARIABLES:
C
C OUTPUT VARIABLE:
C       ZULZZ -- INFRA-RED ABSORPTIVITY OF THE HOT LAYER.
C
C       ZULZZ1 = 0.0005 * EXP(0.0286*ZTZZZ)
C
C A BETTER FIT IS GIVEN BY THE FOLLOWING EQUATION
C
      ZULZZ1 = 0.009464D0 * EXP(0.01842D0*ZTZZZ)
C
C LIMITS:
C       ZULZZ1 = MAX(ZULZZ1,.0005D0)
      RETURN
      END
CHECKED
      SUBROUTINE ABSRB2(ZYLSZ,ZULZZ1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C THIS SUBROUTINE CALCULATES THE INFRA-RED ABSORPTIVITY OF THE HOT LAYER
C WE ASSUME THE ABSORPTION COEFFICIENT IS IN FACT GREY, AND DUE MOSTLY T
C THE SMOKE CONCENTRATION IN THE LAYER.  THE PROPORTIONALITY CONSTANT
C USED IN THE EQUATION BELOW WAS OBTAINED BY COMPARING THE I.R.
C ABSORPTIVITY AS MEASURED IN THE JULY 6, 1977 FULL-SCALE FIRE
C TEST (SEE CHANNEL 126), WITH THE SMOKE CONCENTRATION CALCULATION
C CALCULATED BY LAYR03 IN A STANDARD RUN.
C
C EQUATIONS BY H. MITLER, JULY 1979
C CODED BY B. LONDON, DECEMBER 1979
C ADAPTED TO NEW PHYSICAL SUBROUTINE FORMAT BY J. GAHM, JULY 1981.
C
C INPUT PARAMETERS:
```

```
C
C INPUT VARIABLES:
C      ZYLSZ -- MASS CONCENTRATION OF SMOKE IN THE HOT LAYER.
C
C OUTPUT VARIABLE:
C      ZULZZ1 -- INFRA-RED ABSORPTIVITY OF THE HOT LAYER.
C
C
      ZULZZ1 = 265.D0 * ZYLSZ
C
C LIMITS:
C      ZULZZ1 = MAX(ZULZZ1, .0005D0)
C
      RETURN
      END
CHECKED
      SUBROUTINE ABSRB3(ZKLZZ,ZYLDZ,ZYLMZ,ZYLOZ,ZYLSZ,ZULZZ1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALCULATES THE INFRA-RED ABSORPTIVITY OF THE
C HOT LAYER USING THE ACCURATE BUT TIME CONSUMING SUBROUTINE
C "EMSVTY" TO CALCULATE THE EMISSIVITY OF THE HOT LAYER.
C
C EQUATIONS BY H. MITLER, FEBRUARY 1981.
C CODED BY J. GAHM, FEBRUARY 1981.
C ADAPTED TO NEW PHYSICAL SUBROUTINE FORMAT BY J. GAHM, JULY 1981.
C
C INPUT PARAMETERS:
C
C INPUT VARIABLES:
C      ZKLZZ -- TEMPERATURE OF THE HOT LAYER.
C      ZYLDZ -- MASS CONCENTRATION OF CO2 IN THE HOT LAYER.
C      ZYLMZ -- MASS CONCENTRATION OF CO IN THE HOT LAYER.
C      ZYLOZ -- MASS CONCENTRATION OF O2 IN THE HOT LAYER.
C      ZYLSZ -- MASS CONCENTRATION OF SMOKE IN THE HOT LAYER.
C
C OUTPUT VARIABLE:
C      ZULZZ1 -- INFRA-RED ABSORPTIVITY OF THE HOT LAYER.
C
C
C FIND THE ABSORPTION COEFFICIENT OF THE SOOT IN THE HOT LAYER.
C
      ZKLED = 1.127D6*ZYLSZ/ZKLZZ
C
C FIND PARTIAL PRESSURES OF CO2 AND H2O IN THE HOT LAYER.
C
      SUMPRS = ZYLOZ/32.0D0 + ZYLDZ/44.011D0 +
     1         ZYLMZ/28.011D0 + ZYLSZ/18.016D0
C
      SUMMAS = ZYLOZ + ZYLDZ + ZYLMZ + ZYLSZ
      SUM = .035555D0 + SUMPRS - SUMMAS/28.016D0
      PCO2 = ZYLDZ/(44.011D0*SUM)
      PH2O = ZYLSZ/(18.016D0*SUM)
C
C FIND EMISSIVITY OF THE HOT LAYER.
C
      CALL EMSVTY(ZKLED,1.D0,ZKLZZ,PCO2,PH2O,EMISS)
C
C CONVERT THE EMISSIVITY TO ABSORPTIVITY.
C
      ZULZZ1 = -LOG(1.D0-EMISS)
C
C LIMITS:
```

```
C      ZULZZ1 = MAX(ZULZZ1,.0005D0)
C
      RETURN
      END
CHECKED
      SUBROUTINE HEIGHT(CHI,TPSIO,TMOZB,TMOZZ,ZRFZZ,ZHFZZ1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALCULATES THE HEIGHT OF THE FLAME
C OF A BURNING OBJECT.
C
C EQUATIONS BY H. MITLER.  ADAPTED FROM GFIRO3 CODE BY J. GAHM.
C
C INPUT PARAMETERS:
C     CHI -- FRACTION OF THE HEAT OF COMBUSTION RELEASED IN THE
C            FLAME OF THE OBJECT.
C     TPSIO -- INITIAL VALUE FOR THE TANGENT OF THE SEMI-
C              APEX ANGLE OF THE FLAME OF THE OBJECT.
C
C INPUT VARIABLES:
C     TMOZB -- MASS BURNING RATE OF THE FEUL OF THE OBJECT.
C     TMOZZ -- NEGATIVE OF MASS PYROLYSIS RATE OF THE OBJECT.
C     ZRFZZ -- RADIUS OF THE FIRE ON THE OBJECT.
C
C OUTPUT VARIABLE:
C     ZHFZZ1 -- HEIGHT OF THE FLAMES OVER THE OBJECT.
C
      IF (ABS(TMOZB) .GT. 1.D-15) THEN
        ZHFZZ1 = ZRFZZ * TMOZB / (TPSIO * CHI * ABS(TMOZZ))
      ELSE
        ZHFZZ1 = 0.D0
      END IF
C
C LIMIT CHECK:
C
C     IF (ZHFZZ1 .LT. 0.0D0) THEN
C       ZHFZZ1 = 0.D0
C     END IF
      RETURN
      END
CHECKED
      SUBROUTINE BURN(CHI,XGAMMA,TMOZZ,TMPZZ,TMOZB1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALCULATES THE MASS RATE OF BURNING OF THE FUEL FROM AN
C OBJECT.
C
C EQUATIONS BY H. MITLER. ADAPTED FROM GFIRO3 CODE BY J. GAHM.
C
C INPUT PARAMETERS:
C     CHI -- FRACTION OF HEAT OF COMBUSTION RELEASED IN THE
C            FLAME OF THE OBJECT.
C     XGAMMA -- AIR/FEUL MASS RATIO IN A FREE BURN OF THE
C               OBJECT.
C INPUT VARIABLES:
C     TMOZZ -- NEGATIVE OF MASS PYROLYSIS RATE OF THE OBJECT.
C     TMPZZ -- MASS FLOW INTO HOT LAYER FROM PLUME OF THE
C              OBJECT.
C OUTPUT VARIABLE:
C     TMOZB1 -- MASS RATE OF BURNING OF FUEL FROM THE OBJECT.
C
      TMOZB1 = MIN(-CHI*TMOZZ, (TMOZZ+TMPZZ)/XGAMMA)
      IF (TMOZB1 .LT. 0.D0) TMOZB1 = -0.01D0 * CHI * TMOZZ
      RETURN
      END
CHECKED
      SUBROUTINE CHEM(CP,QF,QVAP,ZKAZZ,ZKOPY,ZRFZM,
     1                TMOZB,ZHPZZ,TEOZZ1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE RATE OT WHICH CHEMICAL ENERGY
C IS RELEASED FROM A BURNING OBJECT.
C
C EQUATIONS BY H. MITLER. ADAPTED FROM GFIRO3 CODE BY J. GAHM.
C
C INPUT PARAMETERS:
C     CP -- SPECIFIC HEAT OF AMBIENT AIR.
C     QF -- HEAT OF COMBUSTION OF THE OBJECT AS FUEL.
C     QVAP -- HEAT OF VAPORIZATION OF THE OBJECT.
C     ZKAZZ -- AMBIENT TEMPERATURE OF THE ROOM CONTAINING
C              THE OBJECT.
C     ZKOPY -- TEMPERATURE OF THE ONSET OF PYROLYSIS OF THE OBJECT.
C     ZRFZM -- MAXIMUM BURNING RADIUS OF THE OBJECT.
C
C INPUT VARIABLES:
C     TMOZB --
C     ZHPZZ -- HEIGHT OF THE OBJECT'S PLUME.
C
C OUTPUT VARIABLE:
C     TEOZZ1 -- CHEMICAL ENERGY RELEASE RATE OF THE OBJECT.
C
C
C JAN. 26, 1979. IF THE OBJECT THINKS IT HAS A NEGATIVE PLUME
C HEIGHT SHUT OFF ITS FIRE.
C
      IF (ZHPZZ .LT. 0.D0) THEN
        TEOZZ1 = 0.D0
        GO TO 10
      END IF
C
      TQF = QF - CP*(ZKOPY-ZKAZZ)
C PATCH TO PREVENT TEOZZ FROM STAYING AT 0. AFTER AN OBJECT MOVES OUT
C OF THE HOT LAYER.
      TMB = TMOZB
      IF (TMOZB.LE.0.D0) TMB = 1.D-6
      TEOZZ1 = -TMB * TQF
C
C LIMITS:
C
   10 CONTINUE
C     IF (TEOZZ1 .LE. 0.D0) THEN
C       BNDS = (-6.D6 * ZRFZM**2 / QVAP) * QF
C       IF (TEOZZ1 .LT. BNDS) TEOZZ1 = BNDS
C     ELSE
C       TEOZZ1 = 0.D0
C     END IF
      RETURN
      END
CHECKED
      SUBROUTINE CNVL1(ZLRZX,ZLRZY,AC,AVLSM,FQLWD,WVLS,TELZZ,
     1                 ZELZZ,ZHLZZ,TELZD1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C FINDS THE RATE OF LOSS OF ENERGY FROM THE HOT LAYER IN
C A ROOM DUE TO CONVECTIVE HEATING OF THE WALLS AND CEILING.
C
```

```
C EQUATIONS BY H. EMMONS. ADAPTED FROM MARK5 VERSION BY J. GAHM.
C
C INPUT PARAMETERS:
C     ZLRZX -- LENGTH OF THE ROOM ALONG THE X-AXIS.
C     ZLRZY -- LENGTH OF THE ROOM ALONG THE Y-AXIS.
C
C INPUT VARIABLES:
C     AQ -- CONVECTIVE ENERGY STORED IN A UNIT AREA OF HOT WALL.
C     AVLSM -- TOTAL AREA OF THE VENTS COVERED BY THE HOT LAYER.
C     FQLWD -- CONVECTIVE HEAT FLUX FROM THE LAYER TO THE ADJACENT
C              WALL.
C     WVLS -- TOTAL WIDTH OF VENTS WHICH INTERSECTS THE INTERFACE
C             BETWEEN THE HOT AND COLD LAYERS (I.E. THE AMOUNT BY
C             WHICH THE PERIMITER OF THE WALLS IS DECREASED BY THE
C             VENT HOLES AT THE INTERFACE.)
C     TELZZ -- TIME RATE OF CHANGE OF TOTAL ENERGY OF THE HOT LAYER.
C     ZELZZ -- TOTAL ENERGY OF THE HOT LAYER.
C     ZHLZZ -- DEPTH OF THE HOT LAYER.
C
C OUTPUT VARIABLE:
C     TELZD1 -- RATE OF CHANGE OF LAYER ENERGY DUE TO CONVECTIVE
C               HEATING OF THE WALLS.
C
      AWC = 2.0D0 * ( ZLRZX + ZLRZY ) * ZHLZZ
      A = AWC + ZLRZY*ZLRZX - AVLSM
      TELZD1 = -A * FQLWD
C
      IF ( TELZZ * ZELZZ .GT. 0.D0 ) THEN
        TELZD1 = TELZD1 - 0.5D0*(AWC-WVLS*ZHLZZ)*AQ*TELZZ
     1                  / ZELZZ
      END IF
C
C LIMITS:
C     BNDS = 2.D6 * ZLRZX * ZLRZY
C     IF ( TELZD1 .GT. BNDS ) THEN
C       TELZD1 = BNDS
C     ELSE IF ( TELZD1 .LT. -BNDS ) THEN
C       TELZD1 = -BNDS
C     END IF
      RETURN
      END
CHECKED
      SUBROUTINE CNVW1(ZOWZM,ZOWZN,ZKLZZ,ZKWZZ,FQLWD1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALCULATES THE CONVECTIVE HEAT TRANSFER TO ONE SIDE OF A WALL
C FROM THE ADJACENT HOT LAYER.
C
C EQUATIONS BY H. EMMONS
C CODED BY L. TREFETHEN, AUGUST 1977
C ADAPTED TO NEW PHYSICAL SUBROUTINE FORMAT BY J. GAHM, AUGUST 1981.
C
C INPUT PARAMETERS:
C     ZOWZM -- MAXIMUM HEAT TRANSFER COEFFICIENT.
C     ZOWZN -- MINIMUM HEAT TRANSFER COEFFICIENT.
C
C INPUT VARIABLES:
C     ZKLZZ -- TEMPERATURE OF THE HOT LAYER.
C     ZKWZZ -- TEMPERATURE OF THE WALL ADJACENT TO THE HOT LAYER.
C
C OUTPUT VARIABLE:
C     FQLWD1 -- CONVECTIVE HEAT FLUX TO THE HOT LAYER FROM THE
C               SURROUNDING WALLS.
```

```
C
      H = MIN( ZOWZM, ZOWZN + (ZOWZM-ZOWZN) * (ZKLZZ-300.D0)/100.D0)
      FQLWD1 = H * ( ZKLZZ - ZKWZZ )
C
C LIMITS:
C     IF (FQLWD1 .GT. 2.D6) THEN
C       FQLWD1 = 2.D6
C     ELSE IF (FQLWD1 .LT. -2.D6) THEN
C       FQLWD1 = -2.D6
C     END IF
C
      RETURN
      END
CHECKED
      SUBROUTINE CVWID(ZBVZZ,ZHTZZ,ZHVZZ,ZHLZZ,WVL1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C THIS SUBROUTINE CALCULATES THE WIDTH OF THE SEGMENT OF A GIVEN
C VENT WHICH INTERSECTS THE INTERFACE BETWEEN THE HOT AND COLD
C LAYERS.  THUS, IF THE LAYER INTERFACE INTERSECTS THE VENT, THE
C SUBROUTINE RETURNS ZBVZZ, AND IF THE INTERFACE DOES NOT INTERSECT
C THE VENT THE SUBROUTINE RETURNS 0.  WHEN THE INTERFACE IS VERY
C CLOSE TO THE TOP OF THE VENT, THE SUBROUTINE RETURNS AN INTER-
C MEDIATE VALUE IN ORDER TO MAINTAIN CONTINUITY.
C
C ADAPTED FROM CNVL CODE BY J. GAHM, AUGUST 1981.
C
C INPUT PARAMETERS:
C     ZBVZZ -- WIDTH OF THE VENT.
C     ZHTZZ -- DISTANCE FROM THE CEILING TO THE TRANSOM OF THE VENT.
C     ZHVZZ -- HEIGHT OF THE VENT.
C
C INPUT VARIABLES:
C     ZHLZZ -- DEPTH OF THE HOT LAYER.
C
C OUTPUT VARIABLE:
C     WVL1 -- LENGTH OF THE INTERSECTION OF THE VENT WITH THE
C             INTERFACE.
C
      HV = ZHLZZ - ZHTZZ
      TOP = .2D0*ZHVZZ
      BOTTOM = .8D0*ZHVZZ
      IF (HV.LT.0.D0 .OR. HV.GT.ZHVZZ) THEN
        WVL1 = 0.D0
      ELSE IF (HV.GE.0.D0 .AND. HV.LT.TOP) THEN
        WVL1 = ZBVZZ*HV/TOP
      ELSE IF(HV.GE.TOP .AND. HV.LT.BOTTOM) THEN
        WVL1 = ZBVZZ
      ELSE IF(HV.GE.BOTTOM .AND. HV.LE.ZHVZZ) THEN
        WVL1 = -ZBVZZ/TOP*(HV-ZHVZZ)
      END IF
      RETURN
      END
CHECKED
      SUBROUTINE DXDCNC(ZMOL,ZMLZZ,ZYLDZ1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE MASS CONCENTRATION OF CARBON
C DIOXIDE IN THE HOT LAYER.
C
C EQS. BY H. EMMONS.  ADAPTED FROM LAYR04 CODE BY J. GAHM,
C JULY 1981.
C
```

```
C INPUT PARAMETERS:
C
C INPUT VARIABLES:
C      ZMDL -- MASS OF CARBON DIOXIDE IN THE HOT LAYER.
C      ZMLZZ -- TOTAL MASS OF THE HOT LAYER.
C
C OUTPUT VARIABLE:
C      ZYLDZ1 -- MASS CONCENTRATION OF CARBON DIOXIDE
C                IN THE HOT LAYER.
C
       PARAMETER (YCO2 = 5.D-4)
C
       IF (ZMDL.LE.ZMLZZ .AND. ZMLZZ.NE.0.D0) THEN
         ZYLDZ1 = ZMDL / ZMLZZ
       ELSE
         ZYLDZ1 = YCO2
       END IF
C
C LIMITS:
C      IF (ZYLDZ1 .LT. YCO2) THEN
C        ZYLDZ1 = YCO2
C      ELSE IF ( ZYLDZ1 .GT. 1.D0 ) THEN
C        ZYLDZ1 = 1.D0
C      END IF
C
       RETURN
       END
CHECKED
       SUBROUTINE DXDPRD(FCO2,TMOZZ,TMOD1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE RATE AT WHICH THE FIRE OF AN OBJECT
C PRODUCES CARBON DIOXIDE.
C
C ADAPTED FROM LAYR04 CODE BY J. GAHM, AUGUST 1981.
C
C INPUT PARAMETERS:
C      FCO2 -- MASS FRACTION OF CO2 IN THE GASES EVOLVED FROM
C              THE PYROLIZING OBJECT.
C
C INPUT VARIABLES:
C      TMOZZ -- NEGATIVE OF MASS PYROLYSIS RATE OF THE OBJECT.
C
C OUTPUT VARIABLE:
C      TMOD1 -- MASS RATE OF EVOLUTION OF CO2 FROM THE OBJECT.
C
       TMOD1 = -FCO2 * TMOZZ
C
       RETURN
       END
CHECKED
       SUBROUTINE ENTPL(TMPLU1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C CALCULATES THE RATE OF ENTRAINMENT OF LAYER GASES
C INTO THE UPPER PART OF THE PLUME OVER AN OBJECT.
C
C EQUATIONS BY H. EMMONS. ADAPTED FROM PLUM CODE BY J. GAHM, JULY 1981.
C
C INPUT PARAMETERS:
C
C INPUT VARIABLES:
C
```

```
C OUTPUT VARIABLE:
C      TMPLU1 -- RATE OF ENTRAINMENT OF LAYER GASES INTO THE UPPER PART
C                OF THE PLUME OVER THE OBJECT.
C
C (6/21/78) THERE IS NO ENTRAINMENT OF VITIATED AIR
C INTO THE PLUME, AS THE PLUME IS (IN THIS VERSION) ASSUMED
C TO TERMINATE AT THE LAYER INTERFACE.  HENCE:
C
      TMPLU1=0.D0
C
C LIMITS:
C      IF (TMPLU1 .NE. 0.D0) TMPLU1 = 0.D0
C
      RETURN
      END
C DEBUGGED AS OF 8/24/80.
      SUBROUTINE FLOW(CD,CP,VMAZZ,ZBVZZ,ZHRZZ,ZHTZZ,ZHVZZ,
     1      ZKAZZI,ZKAZZO,ZHLZZI,ZHLZZO,ZKLZZI,ZKLZZO,ZPRZZI,
     2      ZPRZZO,TEUZZ1,TMDZZ1,TMUZZ1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALCULATES ENERGY AND MASS FLOW THROUGH A VENT.
C THE EQUATIONS WERE DERIVED BY ASSUMING BERNOULLIS EQUATIONS APPLY
C IN AN APPROXIMATE WAY ALONG IMAGINED STREAM LINES AND THE MOMENTUM
C EQUATION APPLIES IN AN APPROXIMATE WAY ALONG HORIZONTAL ELEMENTS.
C THE PRESSURE DIFFERENCE ACROSS A VENT IS A PIECE-WISE LINEAR FUNCTION
C OF HEIGHT AND THE TEMPERATURE AND DENSITY ARE PIECE-WISE CONSTANT
C FUNCTIONS OF HEIGHT. VENT FLOWS ARE CALCULATED BY DIVIDING A VENT
C INTO SECTIONS SUCH THAT THE DENSITY OF AIR IS CONSTANT ON EACH SIDE
C OF THE VENT AND THE SIGN OF THE PRESSURE DIFFERENCE IS THE SAME.
C GIVEN THE PRESSURE DIFFERENCE AT THE BOTTOM OF THE SECTION, ONE CAN
C USE THE FORMULAS DEVELOPED IN THE PAPER BY EMMONS, MACARTHUR, AND
C PAPE, "THE STATUS OF FIRE MODELING IN THE UNITED STATES", TO
C FIND THE PRESSURE AT THE TOP OF THE SECTION AND FINALLY THE MASS
C FLOW THROUGH THE SECTION.
C NOTE: THE PRESSURE IS MEASURED IN METERS OF ATMOSPHERIC AIR.
C
C CODED BY JOHN RAMSDELL.  ADAPTED TO CFC VI FORMAT BY J. GAHM.
C
C INPUT PARAMETERS:
C      CD -- VENT FLOW COEFFICIENT.
C      CP -- SPECIFIC HEAT OF AIR.
C      VMAZZ -- DENSITY OF AMBIENT AIR.
C      ZBVZZ -- WIDTH OF THE VENT.
C      ZHRZZ -- HEIGHT OF THE ROOM(S) CONTAINING THE VENT (NOTE THAT
C               WE ASSUME THAT ANY TWO ROOMS WHICH ARE JOINED BY A
C               VENT HAVE THE SAME HEIGHT.)
C      ZHTZZ -- DISTANCE FROM THE CEILING TO THE TRANSOM OF THE VENT.
C      ZHVZZ -- HEIGHT OF THE VENT.
C      ZKAZZI -- AMBIENT TEMPERATURE OF THE INNER ROOM.
C      ZKAZZO -- AMBIENT TEMPERATURE OF THE OUTER ROOM.
C
C INPUT VARIABLES:
C      ZHLZZI -- DEPTH OF THE LAYER IN THE INNER ROOM (ROOM ON SIDE 1 OF
C                THE VENT).
C      ZHLZZO -- DEPTH OF THE LAYER IN THE OUTER ROOM (ROOM ON SIDE 2 OF
C                THE VENT).
C      ZKLZZI -- TEMPERATURE OF THE LAYER IN THE INNER ROOM.
C      ZKLZZO -- TEMPERATURE OF THE LAYER IN THE OUTER ROOM.
C      ZPRZZI -- ATMOSPHERIC PRESSURE AT THE FLOOR OF THE INNER ROOM.
C      ZPRZZO -- ATMOSPHERIC PRESSURE AT THE FLOOR OF THE OUTER ROOM.
C
C OUTPUT VARIABLES:
```

```
C     TEUZZ1 -- ENERGY OUTFLOW FROM THE HOT LAYER OF THE INNER ROOM
C               THROUGH THE VENT.
C     TMDZZ1 -- MASS OUTFLOW THROUGH THE VENT FROM THE COLD LAYER OF
C               THE INNER ROOM INTO THE COLD LAYER OF THE OUTER ROOM.
C     TMUZZ1 -- MASS OUTFLOW THROUGH THE VENT FROM THE HOT LAYER OF
C               THE INNER ROOM TO THE HOT LAYER OF THE OUTER ROOM.
C
      PARAMETER ( G = 9.8D0 )
C
      TEU = 0.0D0
      TMU = 0.0D0
      TMD = 0.0D0
      R1 = 1.0D0
      R2 = 1.0D0
C
C HT IS THE DISTANCE FROM THE FLOOR TO THE TRANSOM OF THE VENT.
C HV IS THE DISTANCE FROM THE FLOOR TO THE BOTTOM OF THE VENT.
      HT = ZHRZZ - ZHTZZ
      HV = HT - ZHVZZ
C
C HLI IS THE DISTANCE FROM THE FLOOR TO THE BOTTOM OF THE HOT LAYER
C IN THE INNER ROOM.  HLO IS THE CORRESPONDING DISTANCE IN THE
C CORRESPONDING DISTANCE FOR THE OUTER ROOM.
      HLI = ZHRZZ - ZHLZZI
      HLO = ZHRZZ - ZHLZZO
      DELTAP = ZPRZZI - ZPRZZO
      RU1 = ZKAZZI / ZKLZZI
      RU2 = ZKAZZO / ZKLZZO
      TOP = 0.0D0
      PSQ1 = SQRT(ABS(DELTAP))
C BEGIN MAIN LOOP
   10 BOTTOM = TOP
      TOP = HT
      IF (HV .GT. BOTTOM) TOP = HV
      IF (HLI .GT. BOTTOM .AND. TOP .GT. HLI) TOP = HLI
      IF (HLO .GT. BOTTOM .AND. TOP .GT. HLO) TOP = HLO
C TOP AND BOTTOM DELINEARATE THE SEGMENT OF THE VENT OVER
C WHICH THE FLOW IS TO BE CALLLATED
      IF (TOP .GT. HLI) R1 = RU1
      IF (TOP .GT. HLO) R2 = RU2
      HOLDP = DELTAP
      DELTAP = DELTAP + (R2 - R1) * (TOP - BOTTOM)
      IF (DELTAP * HOLDP .LT. 0.D0) THEN
C CHANGE IN DIRECTION OF FLOW DETECTED.
C SEPARATE CALCULATION INTO TWO PARTS.
        TOP = BOTTOM - HOLDP / (R2 - R1)
        DELTAP = 0.0D0
      ENDIF
      PSQ2 = PSQ1
      PSQ1 = SQRT(ABS(DELTAP))
      IF (TOP .LE. HV) GOTO 10
      IF (PSQ1 .GT. 0.0D0 .OR. PSQ2 .GT. 0.0D0) THEN
        TM = (TOP - BOTTOM) *
     1    ROUND((ABS(DELTAP + HOLDP) + PSQ1 * PSQ2) / (PSQ1 + PSQ2))
C FUNCTION ROUND IS LINEAR FOR LARGE VALUES OF X AND IT ROUNDS OUT
C THE SQRT(X) SINGULARITY AT SMALL VALUES OF X. (JDR - 1/82)
        IF (DELTAP + HOLDP .LT. 0.0D0) THEN
C FLOW GOES FROM ROOM 2 TO ROOM 1.
          TM = TM * SQRT(R2)
          IF (TOP .LE. HLO) THEN
            TMD = TMD + TM
          ELSE
            TMU = TMU + TM
            TEU = TEU + TM * ZKLZZO
          ENDIF
        ELSE
          TM = TM * SQRT(R1)
C FLOW GOES FROM ROOM 1 TO ROOM 2.
          IF (TOP .LE. HLI) THEN
            TMD = TMD - TM
          ELSE
            TMU = TMU - TM
            TEU = TEU - TM * ZKLZZI
          ENDIF
        ENDIF
      ENDIF
      IF (TOP .LT. HT) GOTO 10
C END OF MAIN LOOP.
C       0.942809 = SQRT(8.0)/3.0
      TEMP = VMAZZ * 0.942809D0 * CD * ZBVZZ * SQRT(G)
      TEUZZ1 = -CP * TEMP * TEU
      TMUZZ1 = -TEMP * TMU
      TMDZZ1 = -TEMP * TMD
      RETURN
      END
      REAL*8 FUNCTION ROUND(X)
      IMPLICIT INTEGER (A-Z)
      REAL*8 B,X,Y
      PARAMETER (B = 1.0D-3)
C ROUND(X) = X * TANH(ABS(X/(2*B)))
C FOR LARGE X ROUND(X) = X.
C FOR SMALL X ROUND(X) = X * ABS(X) / (2*B)
C JDR - 1/82
      ROUND = X
      Y = ABS(ROUND/B)
      IF (Y .GT. 10.0D0) RETURN
      Y = EXP(-Y)
      ROUND = ROUND * (1.0D0 - Y) / (1.0D0 + Y)
      RETURN
      END
CHECKED


      SUBROUTINE FORV(CD,CP,VMAZZ,ZBVZZ,ZHRZZ,ZHTZZ,ZHVZZ,
     1  ZKAZZI,ZKAZZO,ZHLZZI,ZHLZZO,ZKLZZI,ZKLZZO,ZPRZZI,
     2  ZPRZZO,TEUZZ1,TMDZZ1,ZMUZZ1)
      IMPLICIT REAL*8 (A-H,O-Z)

C   WHERE IS THIS ROUTINE??? 2/6/83

        RETURN
        END


C
      SUBROUTINE FLUX(QVAP,ZUFZZ,FQLOR,FQPOR,FQWOR,ZMOZZ,ZRFZZ,PHI1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUINTE TO CALCULATE THE FLUX INCIDENT ON THE SURFACE OF A
C BURNING TARGET OBJECT.
C
C PART OF GFIRO3 IN CFC MARK5.
C EQUATIONS BY H. MITLER. ADAPTED FROM GFIRO3 CODE BY J. GAHM.
C
C INPUT PARAMETERS:
C      QVAP -- HEAT OF VAPORIZATION OF THE OBJECT.\
C      ZUFZZ -- ABSORPTIVITY OF THE FLAME.
C
C INPUT VARIABLES:
C      FQLOR -- RADIATIVE FLUX FROM THE HOT LAYER TO THE SURFACE OF
C               THE OBJECT.
C      FQPOR -- SUM OF THE RADIATIVE FLUXES FROM THE PLUMES OVER ALL
C               BURNING OBJECTS TO THE SURFACE OF THE TARGET OBJECT.
C      FQWOR -- RADIATIVE FLUX FROM THE WALLS TO THE TARGET OBJECT.
C      ZMCZZ -- MASS OF THE TARGET OBJECT.
C      ZRFZZ -- RADIUS OF THE FIRE ON THE TARGET OBJECT.
C
C OUTPUT VARIABLE:
C      PHI1 -- FLUX INCIDENT ON THE SURFACE OF THE TARGET OBJECT.
C
       PARAMETER (PI = 3.1415927D0)
C
C FIND RADIATIVE FLUX IMPINGING ON TARGET OBJECT FROM ALL HOT SOURCES.
       FQCO = FQPOR + FQLOR + FQWOR
C
C EFFECTIVE RERADIATED FLUX
       FQRR = MIN(13200.D0,21700.D0*ZRFZZ*ZUFZZ)
C
C NET FLUX INCIDENT ON FIRE WHEN FLAME HEIGHT NOT SIGNIFICANTLY
C REDUCED
       FQNET = FQCO - FQRR
C
C MINIMUM FLUX INCIDENT ON FIRE WHEN NOT NEAR BURNOUT
       FQPLUS = FQLOR + FQWOR
C
       FQMAX = MAX(FQNET,FQPLUS)
C
C EFFECTIVE FLUX SEEN BY (FICTITIOUS) ENTIRE SURFACE WHEN FUEL NEARLY
C EXHAUSTED
       IF (ZRFZZ .GT. 0.1D0) THEN
        FQEFF = ZMCZZ*QVAP/(2.D0*PI*ZRFZZ**2)
       ELSE
        FQEFF = FQMAX
       END IF
C
C FLUX INCIDENT ON SURFACE OF FIRE
       PHI1 = MIN(FQMAX,FQEFF)
C
       RETURN
       END
CHECKED
       SUBROUTINE LAYDPT(CP,VMAZZ,ZHRZZ,ZKAZZ,ZLRZX,ZLRZY,
     1          ZELZZ,ZHLZZ1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE DEPTH OF THE LAYER IN A ROOM.
C
C EQUATIONS BY H. EMMONS AND H. MITLER. ADAPTED FROM LAYR04
C CODE BY J. GAHM, AUGUST 1981.
C
C INPUT PARAMETERS:
C      CP -- SPECIFIC HEAT OF AMBIENT AIR.
C      VMAZZ -- DENSITY OF AMBIENT AIR.
C      ZHRZZ -- HEIGHT OF THE ROOM.
C      ZKAZZ -- AMBIENT TEMPERATURE IN THE ROOM.
C      ZLRZX -- LENGTH OF THE ROOM ALONG THE X-AXIS.
C      ZLRZY -- LENGTH OF THE ROOM ALONG THE Y-AXIS.
C
C INPUT VARIABLES:
```

```
C      ZELZZ -- TOTAL ENERGY OF THE LAYER.
C
C OUTPUT VARIABLE:
C      ZHLZZ1 -- DEPTH OF THE LAYER IN THE ROOM.
C
      ZHLZZ1 = ZELZZ / ( ZLRZX * ZLRZY * CP * ZKAZZ * VMAZZ )
C
C LIMITS:
C      IF (ZHLZZ1 .LT. 0.D0) THEN
C        ZHLZZ1 = 0.D0
C      ELSE IF (ZHLZZ1 .GT. ZHRZZ) THEN
C        ZHLZZ1 = ZHRZZ
C      END IF
C
      RETURN
      END
CHECKED
      SUBROUTINE MNXCNC(ZMML,ZMLZZ,ZYLMZ1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE MASS CONCENTRATION OF CARBON
C MONOXIDE IN THE HOT LAYER.
C
C EQS. BY H. EMMONS.  ADAPTED FROM LAYR04 CODE BY J. GAHM,
C JULY 1981.
C
C INPUT PARAMETERS:
C
C INPUT VARIABLES:
C      ZMML -- MASS OF CARBON MONOXIDE IN THE HOT LAYER.
C      ZMLZZ -- TOTAL MASS OF THE HOT LAYER.
C
C OUTPUT VARIABLE:
C      ZYLMZ1 -- MASS CONCENTRATION OF CARBON MONOXIDE
C                IN THE HOT LAYER.
C
      PARAMETER (YCO = 0.D0)
C
      IF (ZMML.LE.ZMLZZ .AND. ZMLZZ.NE.0.D0) THEN
        ZYLMZ1 = ZMML / ZMLZZ
      ELSE
        ZYLMZ1 = YCO
      END IF
C
C LIMITS:
C      IF (ZYLMZ1 .LT. YCO ) THEN
C        ZYLMZ1 = YCO
C      ELSE IF ( ZYLMZ1 .GT. .1D0 ) THEN
C        ZYLMZ1 = .1D0
C      END IF
C
      RETURN
      END
CHECKED
      SUBROUTINE MNXPRD(FCO,TMOZZ,TMOM1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE RATE AT WHICH THE FIRE OF AN OBJECT
C PRODUCES CARBON MONOXIDE.
C
C ADAPTED FROM LAYR04 CODE BY J. GAHM, AUGUST 1981.
C
C INPUT PARAMETERS:
```

```
C      FCC -- MASS FRACTION OF CO EVOLVED IN THE BURNING OF THE OBJECT.
C
C INPUT VARIABLES:
C      TMOZZ -- NEGATIVE OF MASS PYROLYSIS RATE OF THE OBJECT.
C
C OUTPUT VARIABLE:
C      TMCM1 -- MASS RATE OF EVOLUTION OF CO FROM THE OBJECT.
C
       TMCM1 = -FCC * TMOZZ
C
       RETURN
       END
       SUBROUTINE NETMAS(CP,ZKAZZ,TELZZ,TMDSM,TMLZZ,TMOSM,TMUSM,
     1                   TMNET1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C CHECKS FOR CONSERVATION OF MASS IN A GIVEN ROOM.
C
C EQUATIONS BY H. MITLER.  CODED BY J. GAHM, SEPTEMBER 1981.
C
C INPUT PARAMETERS:
C      CP -- SPECIFIC HEAT OF AMBIENT AIR.
C      ZKAZZ -- AMBIENT TEMPERATURE OF THE ROOM.
C
C INPUT VARIABLES:
C      TELZZ -- RATE OF INCREASE OF THE ENERGY OF THE HOT LAYER.
C      TMDSM -- TOTAL MASS OUTFLOW FROM THE COLD LAYER THROUGH
C               ALL THE VENTS IN THE ROOM.
C      TMLZZ -- RATE OF INCREASE OF MASS OF THE HOT LAYER.
C      TMOSM -- NEGATIVE OF SUM OF MASS PYROLYSIS RATES OF ALL
C               OBJECTS IN THE ROOM.
C      TMUSM -- TOTAL MASS OUTFLOW FROM THE HOT LAYER THROUGH ALL
C               THE VENTS IN THE ROOM.
C
C OUTPUT VARIABLE:
C      TMNET1 -- NET RATE OF INCREASE OF MASS WHICH HAS NOT BEEN
C                ACCOUNTED FOR.  NOTE THAT WHEN THE SYSTEM HAS
C                CONVERGED, WE SHOULD HAVE TMNET=0.
C
       TMNET1 = TMLZZ - TELZZ/(CP*ZKAZZ) + TMUSM + TMDSM + TMOSM
C
       RETURN
       END
       SUBROUTINE OXYBRN(XGAMAS,TMOZB,BOXY1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C CALCULATES THE MASS RATE OF OXYGEN BURNING IN THE PLUME OF AN
C OBJECT.
C
C INPUT PARAMETERS:
C      XGAMAS -- STOICHIOMETRIC AIR/FUEL MASS RATIO FOR THE OBJECT.
C
C INPUT VARIABLES:
C      TMOZB -- MASS RATE OF BURNING OF FUEL FROM THE OBJECT.
C
C OUTPUT VARIABLE:
C      BOXY1 -- MASS RATE OF OXYGEN BURNING IN THE PLUME OF THE
C               OBJECT.
C
       PARAMETER (YO2 = .23180D0)
C YO2 -- MASS CONCENTRATION OF OXYGEN IN AMBIENT AIR.
       BOXY1 = YO2 * XGAMAS * TMOZB
C
```

```
      RETURN
      END
CHECKED
      SUBROUTINE OXYCNC(ZMOL,ZMLZZ,ZYLOZ1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE MASS CONCENTRATION OF OXYGEN IN THE
C HOT LAYER.
C
C EQS. BY H. EMMONS.  ADAPTED FROM LAYR04 CODE BY J. GAHM,
C JULY 1981.
C
C INPUT PARAMETERS:
C
C INPUT VARIABLES:
C     ZMOL -- MASS OF OXYGEN IN THE HOT LAYER.
C     ZMLZZ -- TOTAL MASS OF THE HOT LAYER.
C
C OUTPUT VARIABLE:
C     ZYLOZ1 -- MASS CONCENTRATION OF OXYGEN IN THE HOT LAYER.
C
      PARAMETER (YO2 = .2318D0)
C
      IF (ZMOL.LE.ZMLZZ .AND. ZMLZZ.NE.0.D0) THEN
        ZYLOZ1 = ZMOL / ZMLZZ
      ELSE
        ZYLOZ1 = YO2
      END IF
C
C LIMITS:
C     IF ( ZYLOZ1 .LT. 0.D0 ) THEN
C       ZYLOZ1 = 0.D0
C     ELSE IF ( ZYLOZ1 .GT. 1.01D0*YO2 ) THEN
C       ZYLOZ1 = 1.01D0*YO2
C     END IF
C
      RETURN
      END
CHECKED
      SUBROUTINE PLMHT(ZHOZZ,ZHRZZ,ZHLZZ,ZHPZZ1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE HEIGHT OF THE PLUME OVER AN OBJECT.
C
C EQUATIONS BY H. EMMONS.  ADAPTED FROM PLUM CODE BY J. GAHM,
C AUGUST 1981.
C
C INPUT PARAMETERS:
C     ZHOZZ -- HEIGHT OF THE OBJECT.
C     ZHRZZ -- HEIGHT OF THE ROOM CONTAINING THE OBJECT.
C
C INPUT VARIABLES:
C     ZHLZZ -- DEPTH OF THE HOT LAYER IN THE ROOM CONTAINING THE
C              OBJECT.
C
C OUTPUT VARIABLE:
C     ZHPZZ1 -- HEIGHT OF THE PLUME OVER THE OBJECT.
C
      ZHPZZ1 = ZHRZZ - ZHOZZ - ZHLZZ
C
C LIMITS:
C     IF ( ZHPZZ1 .LT. -ZHOZZ ) THEN
C       ZHPZZ1 = -ZHOZZ
```

```
C      ELSE IF ( ZHPZZ1 .GT. ZHRZZ - ZHOZZ ) THEN
C        ZHPZZ1 = ZHRZZ - ZHOZZ
C      END IF
       RETURN
       END
CHECKED
       SUBROUTINE POWL(ZHRZZ,ZLRZX,ZLRZY,TELZD,TELZR,TEPSM,
      1            TEUSM,ZELZZ,TELZZ1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE RATE OF CHANGE OF THE ENERGY OF
C THE HOT LAYER.
C
C EQUATIONS BY H. EMMONS AND H. MITLER. ADAPTED FROM LAYRC4
C CODE BY J. GAHM, AUGUST 1981.
C
C INPUT PARAMETERS:
C      ZHRZZ -- HEIGHT OF THE ROOM.
C      ZLRZX -- LENGTH OF THE ROOM ALONG THE X-AXIS.
C      ZLRZY -- LENGTH OF THE ROOM ALONG THE Y-AXIS.
C
C INPUT VARIABLES:
C      TELZD -- RATE OF CONVECTIVE HEAT TRANSFER TO THE HOT LAYER
C               FROM THE WALLS IN THE SAME ROOM.
C      TELZR -- RATE OF RADIATIVE HEAT TRANSFER TO THE HOT LAYER
C               FROM ALL HOT SOURCES IN THE SAME ROOM.
C      TEPSM -- TOTAL CONVECTIVE ENERGY OUTPUT OF THE PLUMES IN
C               THE ROOM.
C      TELSM -- TOTAL CONVECTIVE ENERGY OUTFLOW THROUGH THE
C               VENTS IN THE ROOM FROM THE HOT LAYER.
C      ZELZZ -- TOTAL ENERGY OF THE LAYER.
C
C OUTPUT VARIABLE:
C      TELZZ1 -- RATE OF INCREASE OF ENERGY OF THE LAYER.
C
       TELZZ1 = TEPSM - TELSM + TELZD + TELZR
C
C LIMITS:
C      IF (TELZZ1 .LT. -2.D9) THEN
C        TELZZ1 = -2.D9
C      ELSE IF (TELZZ1 .GT. 2.D9) THEN
C        TELZZ1 = 2.D9
C      END IF
C
C SINCE ZELZZ IS THE TIME INTEGRAL OF TELZZ, LIMITS ON ZELZZ PUT
C RESTRICTIONS ON TELZZ.  NAMELY, IF ZELZZ IS AT ITS LOWER LIMIT,
C THEN TELZZ MUST BE >= 0 WHILE IF ZELZZ IS AT ITS UPPER LIMIT,
C THEN TELZZ MUST BE <= 0.
C      IF (ZELZZ .LE. 0.D0) THEN
C        TELZZ1 = MAX(TELZZ1,0.D0)
C      ELSE IF (ZELZZ .GE. 2.D6*ZHRZZ*ZLRZX*ZLRZY) THEN
C        TELZZ1 = MIN(TELZZ1,0.D0)
C      END IF
C
       RETURN
       END
CHECKED
       SUBROUTINE PULRAD(ZRFZM,ZRFZO,ZTIG,ZTZZZ,ZRFZZ1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE RADIUS OF A POOL FIRE.
C THE EXPRESSION USED FOR THE RADIUS OF A POOL FIRE PROVIDES
C FOR A SMOOTH BUILD-UP TO MAXIMUM RADIUS TO IMPROVE STABILITY.
```

```
C
C EQUATIONS BY H. MITLER.  CODED BY J. GAHM, JULY 1981.
C
C INPUT PARAMETERS:
C      ZRFZM -- MAXIMUM BURNING RADIUS OF THE OBJECT.
C      ZRFZO -- INITIAL BURNING RADIUS OF THE OBJECT.
C      ZTIG -- TIME OF IGNITION OF THE OBJECT.
C      ZTZZZ -- ABSOLUTE TIME.
C
C INPUT VARIABLES:
C
C OUTPUT VARIABLE:
C      ZRFZZ1 -- RADIUS OF THE POOL FIRE.
C
       ZRFZZ1 = ZRFZM * ( 1.D0 - EXP(-(ZTZZZ-ZTIG)/2.D0) )
C
C LIMITS:
C
C      IF (ZRFZZ1 .LT. ZRFZO) THEN
C        ZRFZZ1 = ZRFZO
C      ELSE
C        IF (ZRFZZ1 .GT. ZRFZM) ZRFZZ1 = ZRFZM
C      END IF
C
       RETURN
       END
CHECKED
       SUBROUTINE PWRPLR(ZLRZX,ZLRZY,TEPZR,ZHPZZ,ZRFZZ,P1PLR1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE POWER ABSORBED BY THE LAYER DUE
C TO RADIATION FROM THE PLUME OF AN OBJECT.
C
C EQUATIONS BY H. MITLER. ADAPTED FROM RDNL CODE BY J. GAHM,
C JULY 1981.
C
C INPUT PARAMETERS:
C      ZLRZX -- LENGTH OF THE ROOM ALONG THE X-AXIS.
C      ZLRZY -- LENGTH OF THE ROOM ALONG THE Y-AXIS.
C
C INPUT VARIABLES:
C      TEPZR -- RADIATIVE POWER OUTPUT FROM THE PLUME OF THE
C               OBJECT.
C      ZHPZZ -- HEIGHT OF THE PLUME OVER THE OBJECT.
C      ZRFZZ -- RADIUS OF THE FIRE ON THE OBJECT.
C
C OUTPUT VARIABLE:
C      P1PLR1 -- POWER FROM THE PLUME OVER THE OBJECT TO THE HOT
C                LAYER IN THE SAME ROOM.
C
       PARAMETER ( PI = 3.1415927D0 )
C
       H = ZHPZZ - ZRFZZ
       P1PLR1 = .5D0 * TEPZR
     1     * ( 1.D0 - H / SQRT(H*H + ZLRZX*ZLRZY/PI) )
       RETURN
       END
CHECKED
       SUBROUTINE PYRO(QVAP,ZRFZM,PHI,ZMOZZ,ZRFZZ,TMOZZ1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE MASS PYROLOSIS RATE OF A BURNING
C OBJECT.
```

```
C
C EQUATIONS BY H. MITLER.  ADAPTED FROM GFIRC3 CODE BY J. GAHM,
C JULY 1981.
C
C INPUT PARAMETERS:
C      QVAP -- HEAT OF VAPORIZATION OF THE OBJECT.
C      ZRFZM -- MAXIMUM BURNING RADIUS OF THE OBJECT.
C
C INPUT VARIABLES:
C      PHI -- FLUX IMPINGING ON THE SURFACE OF THE OBJECT.
C      ZMCZZ -- REMAINING MASS OF THE OBJECT.
C      ZRFZZ -- RADIUS OF THE FIRE ON THE OBJECT.
C
C OUTPUT VARIABLE:
C      TMCZZ1 -- NEGATIVE OF MASS PYROLYSIS RATE OF THE OBJECT.
C
      PARAMETER (PI = 3.1415927D0)
C
      IF (PHI .GT. 0.D0) THEN
        TMO = -PI * ZRFZZ**2 * PHI / QVAP
C
C THE FOLLOWING LINES PRODUCE A SMOOTH CUT-OFF OF BURNING AT
C FUEL EXHAUSTION.
C
        IF (TMO .NE. 0.D0) THEN
          G = ZMCZZ / (2.D0*TMO)
          IF (ABS(G) .LT. 30.D0) THEN
            TMCZZ1 = TMO * (1.D0 - EXP(G))
          ELSE
            TMCZZ1 = TMO
          END IF
        ELSE
          TMCZZ1 = 0.D0
        END IF
      ELSE
        TMCZZ1 = 0.D0
      END IF
C
C LIMITS:
C
C     IF (TMCZZ1 .LE. 0.D0) THEN
C       BNDS = -6.D6 * ZRFZM**2 / QVAP
C       IF (TMOZZ1 .LT. BNDS) TMCZZ1 = BNDS
C     ELSE
C       TMCZZ1 = 0.D0
C     END IF
C
C     IF (ZMCZZ .LE. 0.D0) TMCZZ1 = 0.D0
C
      RETURN
      END
CHECKED
      SUBROUTINE RADFW(ZLRZX,ZLRZY,TEPZR,ZHLZZ,ZHPZZ,ZRFZZ,
     1                 ZULZZ,FQ1PW1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE RADIATIVE FLUX FROM THE PLUME OF
C ONE OBJECT TO THE WALLS IN THE SAME ROOM.
C
C EQUATIONS BY H. MITLER.  ADAPTED FROM RONW CODE BY J. GAHM,
C JUNE 1981.
C
C INPUT PARAMETERS:
```

```
C     ZLRZX -- LENGTH OF ROOM ALONG X-AXIS.
C     ZLRZY -- LENGTH OF ROOM ALONG Y-AXIS.
C
C INPUT VARIABLES:
C     TEPZR -- RATE AT WHICH THE PLUME OF THE OBJECT ADDS ENERGY TO
C              THE HOT LAYER.
C     ZHLZZ -- DEPTH OF THE LAYER.
C     ZHPZZ -- HEIGHT OF THE PLUME OF THE OBJECT.
C     ZRFZZ -- RADIUS OF THE FIRE OF THE OBJECT.
C     ZULZZ -- ABSORPTIVITY OF THE HOT LAYER.
C
C OUTPUT VARIABLE:
C     FQ1PW1 -- RADIATIVE FLUX FROM PLUME OF THE OBJECT TO THE
C               WALLS.
C
      PARAMETER (PI = 3.1415927D0)
C
      WL = ZLRZX * ZLRZY
      AREA = WL + 2.D0 * ( ZLRZX + ZLRZY ) * ZHLZZ
      RCSQ = WL / PI
      B1 = ZULZZ * ZHLZZ
      IF (ZULZZ .NE. 0.D0) THEN
        THETA = MIN( 1.D0/ZULZZ, ZHLZZ/2.D0 )
      ELSE
        THETA = ZHLZZ/2.D0
      END IF
      ZHZ = ZHPZZ - ZRFZZ
C
      IF (ZHZ .GE. 0.D0) THEN
        IF (ZHZ .NE. 0.D0) THEN
          XKI = MIN( ZHZ, THETA )
C
C BELOW, NOTE THAT COS(ATAN(Y))=1./SQRT(1+Y**2)
C
          SECTHT = SQRT( 1.D0 + RCSQ/(ZHZ+XKI)**2 )
          COSTHT = 1.D0/SECTHT
          EXPON = -B1 * SECTHT
          A = COSTHT * ( 1.D0 - EXP(EXPON)/(1.D0-EXPON) )
        ELSE
          A = 0.D0
        END IF
        FQ1PW1 = TEPZR / (2.D0*AREA) * (A + EXP(-B1)/(1.0D0+B1)
     1    - ZHZ/SQRT(RCSQ+ZHZ*ZHZ))
        GO TO 10
C
      ELSE
        CSCTHT = SQRT( 1.D0 + RCSQ/(ZHZ*ZHZ) )
        SINTHT = 1.D0 / CSCTHT
        EXPON = -B1 * CSCTHT
        FQ1PW1 = TEPZR/(2.D0*AREA) * ( EXP(-B1)/(1.D0+B1) +
     1          SINTHT * EXP(EXPON)/(1.D0-EXPON) )
      END IF
C
C LIMITS:
C
   10 CONTINUE
C     IF (FQ1PW1 .LT. 0.D0) THEN
C       FQ1PW1 = 0.D0
C     ELSE
C       IF (FQ1PW1 .GT. 2.D6) FQ1PW1 = 2.D6
C     END IF
C
      RETURN
      END
CHECKED
      SUBROUTINE RADLW(ZLRZX,ZLRZY,ZHLZZ,ZKLZZ,ZULZZ,FQLWR1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE RADIATIVE FLUX FROM THE HOT LAYER
C TO THE PORTION OF THE WALLS AND CEILING COVERED BY THE HOT
C LAYER.
C
C EQUATIONS BY H. MITLER.  ADAPTED FROM RDNW CODE BY J. GAHM,
C JUNE 1981.
C
C INPUT PARAMETERS:
C     ZLRZX -- LENGTH OF THE ROOM ALONG THE X-AXIS.
C     ZLRZY -- LENGTH OF THE ROOM ALONG THE Y-AXIS.
C
C INPUT VARIABLES:
C     ZHLZZ -- DEPTH OF THE HOT LAYER IN THE ROOM.
C     ZKLZZ -- TEMPERATURE OF THE HOT LAYER IN THE ROOM.
C     ZULZZ -- ABSORPTIVITY OF THE HOT LAYER IN THE ROOM.
C
C OUTPUT VARIABLE:
C     FQLWR1 -- RADIATIVE FLUX FROM THE HOT LAYER IN THE ROOM
C               TO THE PORTION OF THE WALLS AND CEILING COVERED
C               BY THE HOT LAYER.
C
      PARAMETER (SIGMA = 5.67D-8)
C
      CEILNG = ZLRZX * ZLRZY
      CVAREA = CEILNG + 2.D0*ZHLZZ*(ZLRZX+ZLRZY)
      VOLUME = ZLRZX * ZLRZY * ZHLZZ
      BOPAC = 4.D0 * ZULZZ * VOLUME / (CVAREA+CEILNG)
      EMITTL = 1.D0 - EXP( -BOPAC/(1.D0 + 0.18D0*BOPAC) )
C
      FQLWR1 = SIGMA * EMITTL * ZKLZZ**4
C
C LIMITS:
C
C     IF (FQLWR1 .LT. 0.D0) THEN
C       FQLWR1 = 0.D0
C     ELSE
C       IF (FQLWR1 .GT. 2.D6) FQLWR1 = 2.D6
C     END IF
C
      RETURN
      END
CHECKED
      SUBROUTINE RDNL(ZKAZZ,ZLRZX,ZLRZY,AVLSM,FQPWR,PPLR,
     1               ZHLZZ,ZKLZZ,ZKWZZ,ZULZZ,TELZR1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALCULATES THE ENERGY LOSSES AND GAINS OF THE HOT LAYER DUE TO
C RADIATION USING AN EMPIRICAL VALUE OF OPTICAL DEPTH (REFER TO
C PAGE 55, CFC III.)
C
C EQUATIONS BY H MITLER
C CODED BY M. WINSLETT, B. LONDON, JUNE 1978
C CONVERTED TO NEW FORMAT BY J. GAHM,  JULY 1981.
C
C INPUT PARAMETERS:
C     ZKAZZ -- AMBIENT TEMPERATURE IN THE ROOM.
C     ZLRZX -- LENGTH OF THE ROOM ALONG THE X-AXIS.
C     ZLRZY -- LENGTH OF THE ROOM ALONG THE Y-AXIS.
```

```
C
C INPUT VARIABLES:
C     AVLSM -- TOTAL AREA OF THE VENTS IN THE ROOM COVERED BY THE
C              HOT LAYER.
C     FQPWR -- RADIATIVE FLUX FROM ALL THE PLUMES IN THE ROOM TO THE
C              WALLS IN THE ROOM.
C     PPLR -- TOTAL RADIATIVE POWER FROM THE PLUMES IN THE ROOM TO
C             THE HOT LAYER.
C     ZHLZZ -- DEPTH OF THE HOT LAYER.
C     ZKLZZ -- TEMPERATURE OF THE HOT LAYER.
C     ZKWZZ -- TEMPERATURE OF THE WALLS IN THE ROOM.
C     ZULZZ -- INFRA-RED ABSORPTIVITY OF THE HOT LAYER.
C
C OUTPUT VARIABLE:
C     TELZR1 -- TOTAL RADIATIVE POWER TO THE HOT LAYER.
C
      PARAMETER ( PI = 3.1415927D0, SIGMA = 5.67D-8 )
C
C CEILING AREA
      WL = ZLRZX * ZLRZY
C
C VOLUME OF HOT LAYER:
      V = WL * ZHLZZ
C
C BOUNDING AREA OF HOT LAYER
      A = 2.D0 * ( WL + (ZLRZY + ZLRZX) * ZHLZZ )
C
C MEAN EMISSIVITY OF LAYER: VARIABLE BEAM LENGTH
C APPROXIMATION (REFER TO PAGE 756, DR. MITLER'S NOTES)
      ZED = 4.D0 * ZULZZ * V / A
      EBAR=1.D0-EXP(-ZED/(1.D0+.18D0*ZED))
C
C RADIATIVE HEAT GAIN RATE OF HOT LAYER, EXCLUDING FLAMES
      DUMMY=A*ZKLZZ**4-(WL+AVLSM)*ZKAZZ**4
      DUMMY=DUMMY-(A-WL-AVLSM)*ZKWZZ**4
      QDOTLR=-SIGMA*EBAR*DUMMY
C
C THE FLAME RADIATION ABSORBED BY THE CEILING
C OR GOING OUT THE VENTS IS:
      EDOTFC=FQPWR*(A-WL)
C
C THE PART OF THE FLAME RADIATION ABSORBED BY THE LAYER IS:
      EDOTA=PPLR-EDOTFC
C
      TELZR1=QDOTLR+EDOTA
C
C LIMITS:
C     BNDS = 2.D6 * ZLRZY * ZLRZX
C     IF (TELZR1 .GT. BNDS) THEN
C       TELZR1 = BNDS
C     ELSE IF (TELZR1 .LT. -BNDS) THEN
C       TELZR1 = -BNDS
C     END IF
C
      RETURN
      END
C NOT CHECKED
      SUBROUTINE RDNP(ZUFZZ,T,ZTZZZ,KO,ZRFZZ,ZHFZZP,
     1   ZLRZX,ZLRZY,ZHFZZ,ZMCZZ,TMOP,TMOZZ,TEOZZ,TEPZR1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE CALCULATES TOTAL RADIATIVE HEAT LOSS (TEPZR1)
C FROM THE FLAME OF AN OBJECT.  THE FLAME IS ASSUMED TO BE
```

```
C A CONE OR CONICAL SHELL.
C
C EQUATIONS BY H. MITLER.
C ADAPTED FROM THE MARK5 VERSION BY J. GAHM AND B.J. MCLAUGHLIN.
C JULY 1981.
C
C INPUT PARAMETERS:
C      ZLRZX -- LENGTH OF ROOM ALONG X-AXIS.
C      ZLRZY -- LENGTH OF ROOM ALONG Y-AXIX.
C      KO -- OBJECT NUMBER OF THE OBJECT.
C      ZHFZZP -- HEIGHT OF THE FLAMES OVER THE OBJECT AT THE PREVIOUS
C                TIME STEP.
C      ZTZZZ -- TIME.
C      T -- TEMPERATURE OF THE FLAME OF THE OBJECT.
C      TMOP -- NEGATIVE OF THE MASS PYROLYSIS RATE OF THE OBJECT
C              AT THE PREVIOUS TIME STEP.
C      ZUFZZ -- ABSORPTIVITY OF THE FLAME OF THE OBJECT.
C
C INPUT VARIABLES:
C      TEOZZ -- RATE AT WHICH THE FIRE OF THE OBJECT PRODUCES
C               CHEMICAL ENERGY.
C      TMOZZ -- NEGATIVE OF MASS PYROLYSIS RATE OF THE OBJECT.
C      ZRFZZ -- RADIUS OF THE FIRE ON THE OBJECT.
C      ZHFZZ -- HEIGHT OF THE FLAMES OVER THE OBJECT.
C      ZMOZZ -- MASS OF OBJECT.
C
C OUTPUT VARIABLE:
C      TEPZR1 -- RATE OF HEAT LOSS FROM THE FLAME OF THE OBJECT VIA
C                RADIATION.
C
C NOTE THAT THE FOLLOWING INCLUDE FILE IS NEEDED ONLY TO GET THE
C MAXIMUM NUMBER OF OBJECTS IN THE RUN FOR THE DIMENSIONS OF ARRAYS.
$INCLUDE 9/DIMN.CMN
      PARAMETER (MO1 = MO*MR)
      REAL*8 PI,SIGMA,ALEPH
      PARAMETER (PI = 3.1415927D0, SIGMA = 5.67D-8, ALEPH = 0.0D0)
      LOGICAL SET(MO1)
      REAL*8 ALPHAO(MO1),A,BNDS,CO,H,R,SI,TEPMAX,TP,V,XI,ZETA
      SAVE SET,ALPHAO
      DATA SET / MO1 * .TRUE./
      IF (ZMOZZ .LE. 0.0D0) THEN
        TEPZR1 = 0.0D0
        RETURN
      END IF
      HYP = SQRT(ZHFZZ*ZHFZZ + ZRFZZ*ZRFZZ)
      IF (HYP .NE. 0.D0) THEN
C      COSINE OF SEMI-APEX ANGLE OF FLAME.
       CO = ZHFZZ / HYP
C      SINE OF SEMI-APEX ANGLE OF FLAME.
       SI = ZRFZZ / HYP
      ELSE
       CO = 0.D0
       SI = 1.D0
      END IF
      IF (ZMOZZ .LT. 0.2D0) THEN
        IF (SET(KO)) THEN
          ALPHAO(KO) = ZHFZZP / SQRT (ABS(TMOP))
          SET(KO) = .FALSE.
        END IF
        H = ALPHAO(KO) * SQRT (ABS(TMOZZ))
        HYP = SQRT (ZRFZZ * ZRFZZ + H * H)
        IF (HYP .NE. 0.D0) THEN
         CO = H / HYP
         SI = ZRFZZ / HYP
        ELSE
         CO = 0.D0
         SI = 1.D0
        END IF
      END IF
      R = ZRFZZ
C     A IS THE AREA OF CONICAL SHELL, V ITS VOLUME.
      IF (SI .EQ. 0.D0) THEN
       A = 0.D0
      ELSE
       A = PI * R**2 * ((1.0D0 + 1.0D0 / SI)*(1.0D0 + ALEPH**2)
     1       - 2.0D0 * ALEPH**2)
      END IF
      IF (SI .NE. 0.D0) THEN
       V = (PI * R**3 / 3.0D0) * (1.0D0 - ALEPH**3) * CO / SI
      ELSE
       V = 0.D0
      END IF
      ZETA = 4.0D0 * V * ZUFZZ / A
      XI = ZETA / (1.0D0 + 0.18D0 * ZETA)
      TEPZR1 = A * SIGMA * T**4 * (1.0D0 - EXP(-XI))
C     LIMITS
      TEPMAX = -0.43D0 * TEOZZ
      IF (ZTZZZ .GT. 2.0D0) THEN
        IF (TEPZR1 .GT. TEPMAX) TEPZR1 = TEPMAX
      END IF
C
C LIMITS:
C     IF (TEPZR1 .LT. 0.0D0) THEN
C       TEPZR1 = 0.0D0
C     ELSE
C       BNDS = 2.0D6 * ZLRZX * ZLRZY
C       IF (TEPZR1 .GT. BNDS) TEPZR1 = BNDS
C     END IF
C
      RETURN
      END
CHECKED
      SUBROUTINE RNFF02(T,ZHOZZF,ZHOZZT,ZHRZZ,ZRFZIT,ZRFZIF,
     1           ZUFZZ,ZXOZZF,ZXOZZT,ZYOZZF,ZYOZZT,ZHFZZF,ZHFZZT,
     2           ZHLZZ,ZRFZZF,ZRFZZT,FQPP1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE RADIATIVE FLUX ABSORBED BY THE FLAME
C OF A BURNING TARGET OBJECT FROM THE FLAME OF A BURNING SOURCE OB-
C JECT.  NOTE THAT THE SUBROUTINE ASSUMES THAT THE OBJECTS DO NOT
C OVERLAP AND THAT THE TARGET AND SOURCE ARE NOT THE SAME OBJECT.
C DIFFERS FROM RNFF01 IN THAT THE CALCULATIONS ARE MUCH
C MORE CRUDE AND SIMPLE.
C
C EQUATIONS BY H. MITLER. ADAPTED FROM THE MARK5 VERSION BY J. GAHM,
C JULY 1981.
C
C INPUT PARAMETERS:
C     T -- TEMPERATURE OF THE FLAMES.
C     ZHOZZF -- HEIGHT OF THE SOURCE OBJECT.
C     ZHOZZT -- HEIGHT OF THE TARGET OBJECT.
C     ZHRZZ -- HEIGHT OF THE ROOM CONTAINING THE TWO OBJECTS.
C     ZRFZIF -- OBJECT RADIUS OF THE SOURCE.
C     ZRFZIT -- OBJECT RADIUS OF THE TARGET.
C     ZUFZZ -- ABSORPTIVITY OF THE FLAMES.
C     ZXOZZF -- X-COORDINATE OF THE SOURCE.
```

```
C       ZXOZZT -- X-COORDINATE OF THE TARGET.
C       ZYOZZF -- Y-COORDINATE OF THE SOURCE.
C       ZYOZZT -- Y-COORDINATE OF THE TARGET.
C
C INPUT VARIABLES:
C       ZHFZZF -- HEIGHT OF THE FLAME OVER THE SOURCE OBJECT.
C       ZHFZZT -- HEIGHT OF THE FLAME OVER THE TARGET OBJECT.
C       ZHLZZ -- DEPTH OF THE LAYER IN THE ROOM CONTAINING THE OBJECTS.
C       ZRFZZF -- RADIUS OF THE FIRE ON THE SOURCE OBJECT.
C       ZRFZZT -- RADIUS OF THE FIRE ON THE TARGET OBJECT.
C
C OUTPUT VARIABLE:
C       FQPP1 -- RADIATIVE FLUX TO THE SURFACE OF THE TARGET TO DUE
C                TO THE ABSORBTION OF FLUX FROM THE SOURCE IN THE FLAME
C                OF THE TARGET.
C
        PARAMETER (PI = 3.1415927D0)
        PARAMETER (SIGMA = 5.67D-8)
        LOGICAL HIGHT
C
C IF EITHER OBJECT IS IN THE HOT LAYER THEN THE FLUX TO THE TARGET
C FLAME IS 0.
        IF ( ZHRZZ-ZHLZZ .LE. MAX(ZHOZZF,ZHOZZT) ) THEN
          FQPP1 = 0.D0
          RETURN
        END IF
C
        D = SQRT( (ZXOZZT-ZXOZZF)**2 + (ZYOZZT-ZYOZZF)**2 )
C
C IF THE OBJECTS OVERLAP THEN ASSUME 0. FLUX TO THE TARGET FLAME.
C NOTE THAT THE FOLLOWING LINE ASSUMES SQUARE OBJECTS.
        IF ( D .LT. .886227D0*(ZRFZIT+ZRFZIF) ) THEN
          FQPP1 = 0.D0
          RETURN
        END IF
C
C SEE IF THE TARGET IS HIGHER THAN THE SOURCE.
        IF ( ZHOZZT .GE. ZHOZZF ) THEN
          HIGHT = .TRUE.
        ELSE
          HIGHT = .FALSE.
        END IF
C
C RENAME VARIABLES SO THAT OBJECT 1 IS THE HIGHER OF SOURCE AND TARGET.
        IF (HIGHT) THEN
          R1 = ZRFZZT
          R2 = ZRFZZF
C ASSUME SQUARE OBJECTS.
          RZ = .886227D0 * ZRFZIT
          IF (1000.D0*ZHFZZT .GT. ZRFZZT) THEN
           TA1 = ZRFZZT/ZHFZZT
          ELSE
           TA1 = 1000.D0
          END IF
          IF (1000.D0*ZHFZZF .GT. ZRFZZT) THEN
           TA2 = ZRFZZF/ZHFZZF
          ELSE
           TA2 = 1000.D0
          END IF
          T1 = T
          T2 = T
          HC1 = ZHOZZT
          HC2 = ZHOZZF
          ZKAP1 = ZUFZZ
          ZKAP2 = ZUFZZ
        ELSE
          R1 = ZRFZZF
          R2 = ZRFZZT
C ASSUME SQUARE OBJECTS.
          RZ = .886227D0 * ZRFZIF
          IF (1000.D0*ZHFZZT .GT. ZRFZZT) THEN
           TA2 = ZRFZZT/ZHFZZT
          ELSE
           TA2 = 1000.D0
          END IF
          IF (1000.D0*ZHFZZF .GT. ZRFZZT) THEN
           TA1 = ZRFZZF/ZHFZZF
          ELSE
           TA1 = 1000.D0
          END IF
          T1 = T
          T2 = T
          HO1 = ZHCZZF
          HO2 = ZHCZZT
          ZKAP1 = ZUFZZ
          ZKAP2 = ZUFZZ
        END IF
C
C GET TRIG FUNCTIONS OF SEMI APEX ANGLES:
        CO1 = 1.D0 / SQRT( 1.D0 + TA1*TA1 )
        SI1 = TA1 * CO1
        COT1 = 1.D0 / TA1
        CSC1 = 1.D0 / SI1
        CO2 = 1.D0 / SQRT(1.D0 + TA2*TA2)
        SI2 = TA2 * CO2
        COT2 = 1.D0 / TA2
        CSC2 = 1.D0 / SI2
C
C AN AD HOC CHANGE ON 8/1/80: R1 MUST BE LESS THAN RZ FOR CONSISTENCY
C OF THIS SUBROUTINE. ZRFZZ IS PRESENTLY DEFINED SO AS TO TAKE INTO
C ACCOUNT BURNING ON THE SIDES OF THE OBJECT, SO IT COULD BE LARGER
C THAN THE OBJECT RADIUS.
        R1 = MIN(R1,RZ)
C
        HI = ZHRZZ - ZHLZZ - HO2
        H1 = R1 * COT1
        H2 = R2 * COT2
        HZ = HO1 -HO2
C
C AMOUNT BY WHICH HEIGHT OF CONE 2 EXCEEDS HEIGHT OF SLAB
C ON WHICH CONE 1 STANDS:
        HU = MAX( H2-HZ, 0.D0 )
        RU = HU *TA2
        GX = 2.D0 * SIGMA * T1**4 * ZKAP1 * ZKAP2 / (D*D)
        HC = HZ * RZ / (D-RZ+R2)
C
C RATIO OF ABSORBING VOLUME OF LOWER PART OF CONE 2 TO ITS BOUNDING
C AREA. 4 TIMES THIS QUANTITY IS (APPROX.) THE MEAN BEAM LENGTH:
        IF (H1 .LT. HC) THEN
          CL = ((HU+(D-RZ)*H1/RZ)**3 - HU**3) /
       1 (3.D0 * ((HU+(D-RZ)*H1/RZ)**2*(1.D0+CSC2)+HU*HU*(1.D0-CSC2)))
        ELSE
          CL = (H2**3 - HU**3) /
       1          (3.D0 * (H2*H2*(1.D0+CSC2)+HU*HU*(1.D0-CSC2)))
        END IF
C
```

```
      QU = HU / (3.D0*(1.D0+CSC2))
C V1 IS PART OF CONE 1 NOT OBSCURED BY THE HOT LAYER.
      IF (HI .LT. HZ) THEN
        V1 = 0.D0
      ELSE
        V1 = R1 * R1 * H1 / 3.D0
        IF (HI .LT. H1+HZ)
     1        V1 = V1 - TA1*TA1*(HZ+H1-HI)**3/3.D0
      END IF
      IF (HU .NE. 0.D0) THEN
        ARG = RU/D
        IF (ABS(ARG) .LE. 1.D0) THEN
          ANGLE = ASIN(ARG)
        ELSE
          ANGLE = .5D0 * PI
        END IF
        AU = 2.D0 * D*D *ANGLE * (SQRT(HU*HU + D*D)-D)/HU
      ELSE
        AU = 0.D0
      END IF
C
C APU IS THE (APPROX.) PROJECTED AREA OF THE UPPER PART OF CONE 2.
C APL - OF THE VISIBLE PART OF THE LOWER SECTION.
      IF (HZ .GE. HI) THEN
        APU = 0.D0
      ELSE IF (HI .LT. H2) THEN
        ARG = (R2 - HI*TA2) / D
        IF (ABS(ARG) .LE. 1.D0) THEN
          ANGLE = ASIN(ARG)
        ELSE
          ANGLE = .5D0 * PI
        END IF
        APU = AU - 2.D0 * D*D * ANGLE * (SQRT((H2-HI)**2+D*D)-D) /
     1           (H2-HI)
      ELSE
        APU = AU
      END IF
C
      IF (H1 .GT. HC) THEN
        ARG = R2 / D
        IF (ABS(ARG) .LE. 1.D0) THEN
          ANGLE = ASIN(ARG)
        ELSE
          ANGLE = .5D0 * PI
        END IF
        APL = 2.D0 * D*D * ANGLE * (SQRT(H2*H2+D*D)-D)/H2 - AU
      ELSE
        APL = (D-RZ) * (2.D0*HU + (D-RZ)*H1/RZ) * H1/RZ * TA2
      END IF
C
C ZM IS (ROUGHLY) THE MEAN FRACTION OF THE EMITTING CONE
C SEEN BY THE LOWER PART OF THE ABSORBING CONE.
      DIFZ = D - RZ - R2
      IF (DIFZ .LE. 0.D0) DIFZ = D * 1.D-3
      ZM = MAX(.5D0 + (1.D0 - RZ*HZ/(H1*DIFZ))**3/2.D0, .5D0)
C
C WE ARRIVE AT THE EXPRESSION FOR THE RATE AT WHICH ENERGY IS
C ABSORBED BY THE LOWER PART OF CONE 2.
C
      ELT = 2.D0* QL * APL
C
C EUT IS THE POWER ABSORBED BY THE PART OF CONE 2 LYING ABOVE THE
C SLAB OF OBJECT 1.
```

```
      EUT = 2.D0 * QU * APU
      E12 = GX * V1 * (EUT + ZM*ELT)
C
      IF (HIGHT) THEN
        E21 = E12 * (T2/T1)**4 * (1.D0+SI2)/(1.D0+SI1) * CO1/CO2
        FQPP1 = E21 / (PI * R1*R1 * (1.D0+CSC1))
      ELSE
        FQPP1 = E12 / (PI*R2*R2*(1.D0+CSC2))
      END IF
      RETURN
      END
CHECKED
      SUBROUTINE RNLH(ISTAT,ZHOZZ,ZHRZZ,ZLRZX,ZLRZY,ZUFZZ,
     1                ZXOZZ,ZYOZZ,ZHFZZ,ZHLZZ,ZKLZZ,ZRFZZ,ZULZZ,
     2                FQLOR1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALCULATES THE RADIATIVE FLUX TO THE HORIZONTAL SURFACE
C OF AN OBJECT FROM THE LAYER IN THE SAME ROOM VIA A CYLINDER APPROX-
C IMATION (EVEN IF THE OBJECT IS BURNING).
C
C EQUATIONS BY H. MITLER.  ADAPTED FROM MARKS CODE BY J. GAHM,
C JULY 1981.
C
C INPUT PARAMETERS:
C      ISTAT -- THE STATE OF THE OBJECT (ISTAT = 5 --> OBJECT IS
C               BURNING.)
C      ZHOZZ -- THE HEIGHT OF THE OBJECT.
C      ZHRZZ -- THE HEIGHT OF THE ROOM CONTAINING THE OBJECT.
C      ZLRZX -- THE LENGTH ALONG THE X-AXIS OF THE ROOM CONTAINING
C               THE OBJECT.
C      ZLRZY -- THE LENGTH ALONG THE Y-AXIS OF THE ROOM CONTAINING
C               THE OBJECT.
C      ZUFZZ -- THE ABSORPTIVITY OF THE FLAMES OF THE OBJECT.
C      ZXOZZ -- THE X-COORDINATE OF THE OBJECT.
C      ZYOZZ -- THE Y-COORDINATE OF THE OBJECT.
C
C INPUT VARIABLES:
C      ZHFZZ -- HEIGHT OF THE FLAMES OVER THE OBJECT.
C      ZHLZZ -- THE DEPTH OF THE LAYER IN THE ROOM CONTAINING THE OBJECT.
C      ZKLZZ -- THE TEMPERATURE OF THE LAYER IN THE ROOM CONTAINING THE
C               OBJECT.
C      ZRFZZ -- THE RADIUS OF THE FIRE ON THE OBJECT.
C      ZULZZ -- THE ABSORPTIVITY OF THE HOT LAYER IN THE ROOM
C               CONTAINING THE OBJECT.
C
C OUTPUT VARIABLE:
C      FQLOR1 -- RADIATIVE FLUX FROM THE HOT LAYER TO THE SURFACE OF
C                THE OBJECT.
C
      PARAMETER (PI = 3.1415927D0)
      PARAMETER (SIGMA = 5.67D-8)
      DIMENSION A(4)
C
      X1 = ZHRZZ - ZHOZZ
      X0 = X1 - ZHLZZ
      A(1) = ZXOZZ * ZYOZZ
      A(2) = (ZLRZX - ZXOZZ) * ZYOZZ
      A(3) = (ZLRZY - ZYOZZ) * ZXOZZ
      A(4) = (ZLRZX - ZXOZZ) * (ZLRZY - ZYOZZ)
C
C SUM E-PRIME OVER THE FOUR AREAS:
      SUM = 0.D0
      DO 10 I=1,4
        R2 = MAX( 4.D0*A(I)/PI, 1.D-10 )
        ZZ0 = SQRT( R2 + X0*X0 )
        ZZ1 = SQRT( R2 + X1*X1 )
        EXP01 = ( 1.D0 + X0*X0/R2 ) * (ZHLZZ - ZZ1 + ZZ0) * ZULZZ
        E0 = 1.D0 - EXP(-2.D0*EXP01)
        SUM = SUM + E0 / (1.D0+X0*X0/R2)
   10 CONTINUE
      FQLOR1 = .25D0 * SIGMA * ZKLZZ**4 * SUM
C
C VERY CRUDE APPROXIMATION TO MEAN OPTICAL DEPTH:
C
C APPROXIMATION IMPROVED ON JUNE 27, 1978: THIS TAKES ACCOUNT OF THE
C HEATING OF THE FLAME BY THE LAYER AND CONSEQUENT INCREASED RADIATION
C TO THE SURFACE.
      IF (ISTAT .EQ. 5) THEN
        TAU = ZRFZZ * ZUFZZ
        HYP = SQRT(ZRFZZ*ZRFZZ + ZHFZZ*ZHFZZ)
        IF (HYP .NE. 0.D0) THEN
         SI = ZRFZZ / HYP
        ELSE
         SI = 1.D0
        END IF
        FQLOR1 = FQLOR1 * (EXP(-TAU) + SI) / (1.D0 + SI)
      END IF
C
C LIMITS:
C     IF (FQLOR1 .LT. 0.D0) THEN
C       FQLOR1 = 0.D0
C     ELSE IF (FQLOR1 .GT. 2.D6) THEN
C       FQLOR1 = 2.D6
C     END IF
C
      RETURN
      END
C NOT DEBUGGED AS OF 9/3/81
      SUBROUTINE RNLV(ANGV,KO,ZHOZZ,ZHRZZ,ZLRZX,ZLRZY,ZXOZZ,ZYOZZ,
     1                   ZHLZZ,ZKLZZ,ZULZZ,FQLOR1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALCULATES THE RADIATIVE FLUX FROM THE LAYER TO
C THE SURFACE OF A VERTICAL OBJECT VIA A CYLINDER APPROXIMATION
C (EVEN IF THE OBJECT IS BURNING).
C
C EQUATIONS BY H. MITLER.
C ADAPTED FROM MARK5 VERSION BY B.J. MCLAUGHLIN
C
C INPUT PARAMETERS:
C     ANGV -- DEVIATION ANGLE FROM PARALLELISM WITH THE WALL
C             RUNNING ALONG THE X-AXIS.
C     KO -- AN IDENTIFIER NUMBER FOR THE OBJECT TO DISTINGUISH
C           IT FROM ALL OTHER OBJECTS.
C     ZHOZZ -- HEIGHT OF THE OBJECT.
C     ZHRZZ -- HEIGHT OF THE ROOM.
C     ZLRZX -- LENGTH OF THE ROOM ALONG THE X-AXIS.
C     ZLRZY -- LENGTH OF THE ROOM ALONG THE Y-AXIS.
C     ZXOZZ -- X-COORDINATE OF THE OBJECT.
C     ZYOZZ -- Y-COORDINATE OF THE OBJECT.
C
C INPUT VARIABLES:
C     ZHLZZ -- DEPTH OF THE HOT LAYER.
C     ZKLZZ -- TEMPERATURE OF THE HOT LAYER.
C     ZULZZ -- INFRA-RED ABSORPTIVITY OF THE HOT LAYER.
```

```
C
C OUTPUT VARIABLE:
C      FQLOR1 -- RADIATIVE FLUX FROM THE HOT LAYER TO THE
C                SURFACE OF THE OBJECT.
C NOTE THAT THE FOLLOWING INCLUDE FILE IS NEEDED ONLY TO GET THE
C MAXIMUM NUMBER OF OBJECTS IN THE RUN FOR THE DIMENSIONS OF VARIOUS
C ARRAYS.
$INCLUDE 9/DIMN.CMN
      PARAMETER (MO1 = MO*MR)
      PARAMETER(PI = 3.14159270D0, SIGMA = 5.67D-8)
      DIMENSION R(2,MO1),ROT1(MO1),MOAR(MO1)
      LOGICAL MOAR
      DATA MOAR /MO1*.FALSE./
      SAVE MOAR,ROT1,R
C
C FIND THE EFFECTIVE RADII R(1,KO), AND R(2,KO) AND THE
C ADJUSTED ANGLE ROT1(KO) ONLY ONCE DURING EACH RUN.
      IF (.NOT.MOAR(KO)) THEN
        MOAR(KO) = .TRUE.
        ICASE = INT(ANGV/90.D0)
        ROT1(KO) = (ANGV-DBLE(ICASE)*90.D0) * (180.D0/PI)
        GO TO (10,20,30,40) ICASE+1
   10   XL = ZLRZX
        YL = ZLRZY
        A = ZXOZZ
        B = ZYOZZ
        GO TO 50
   20   XL = ZLRZY
        YL = ZLRZX
        A = ZYOZZ
        B = ZLRZX-ZXOZZ
        GO TO 50
   30   XL = ZLRZX
        YL = ZLRZY
        A = ZLRZX - ZXOZZ
        B = ZLRZY - ZYOZZ
        GO TO 50
   40   XL = ZLRZY
        YL = ZLRZX
        A = ZLRZY - ZYOZZ
        B = ZXOZZ
   50   CONTINUE
        TROT1 = TAN(ROT1(KO))
        S1 = (XL-A)**2 * TROT1 / 2.D0
        IF (TROT1*(XL-A) .GT. (YL-B))
     1      S1 = (XL-A)*(YL-B) - (YL-B)**2/TROT1/2.D0
        S2 = A*(YL-B) - (YL-B)**2*TROT1/2.D0
        IF (TROT1*(YL-B) .GT. A)  S2 = A*A/TROT1/2.D0
        S3 = A*A*TROT1/2.D0
        IF (TROT1*A .GT. B) S3 = A*B - B*B/TROT1/2.D0
        S5 = XL*(YL-B) - S1 - S2
        R(1,KO) = SQRT(4.D0*(S2+S3)/PI)
        R(2,KO) = SQRT(4.D0*S5/PI)
      END IF
C
C FIND DISTANCE FROM OBJECT TO THE LAYER
      XH = ZHRZZ - ZHLZZ - ZHOZZ
      XO = ABS(XH)
      X1 = ZHLZZ + XH
C IS THE OBJECT IMMERSED IN THE LAYER?
      IF (XH .LE. 0.D0) GO TO 70
C CASE: OBJECT NOT IN THE LAYER.
      SUM = 0.D0
      DO 60 I=1,2
        BET = ATAN(R(I,KO)/X0) -
     1      X0*R(I,KO)/(X0*X0+R(I,KO)*R(I,KO))
        R2 = R(I,KO)*R(I,KO)
        Z0 = SQRT(R2+X0*X0)
        Z1 = SQRT(R2+X1*X1)
        BEAM = (2.D0*PI/BET) *
     1     (X1*LOG((Z1+R(I,KO))/X1) - X0*LOG((Z0+R(I,KO))/X0)   )
   60 SUM = SUM + BET * (1.D0-EXP(-BEAM*ZULZZ))
      FQLOR1 = (SIGMA*ZKLZZ**4/(2.D0*PI)) * SUM
      RETURN
C CASE: OBJECT IMMERSED IN THE LAYER.
   70 SUM = 0.D0
      DO 80 I=1,2
        R2 = R(I,KO)*R(I,KO)
        Z0 = SQRT(R2+X0*X0)
        Z1 = SQRT(R2+X1*X1)
        P0 = 0.D0
        IF (X0 .NE. 0.D0)
     1     P0 = 4.D0 * X0 * LOG( (Z0+R(I,KO))/X0 )
        P1 = 0.D0
        IF (X1 .NE. 0.D0)
     1     P1 = 4.D0 * X1 * LOG( (Z1+R(I,KO))/X1 )
   80 SUM = SUM + 2.D0 - EXP(-ZULZZ*P0) - EXP(-ZULZZ*P1)
      FQLOR1 = (SIGMA * ZKLZZ**4 / 4.D0) * SUM
      RETURN
      END
      SUBROUTINE RNPC01(ISTATT,KOF,T,TMOPF,HFPF,
     1     ZHOZZF,ZHOZZT,ZHRZZ,ZRFZIF,ZRFZIT,ZUFZZ,
     2     ZXOZZF,ZXOZZT,ZYOZZF,ZYOZZT,TMOZZF,ZHFZZF,ZHLZZ,ZHPZZF,
     3     ZMOZZF,ZRFZZF,ZRFZZT,FQ1PC1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALCULATES THE RADIATIVE FLUX FROM THE PLUME OVER
C A BURNING OBJECT (FIRE) TO THE SURFACE OF A TARGET OBJECT.  ONE
C FORMULA IS USED IF THE FLAME IS ASSUMED TO BE A SHELL (ISHELL=1)
C AND ANOTHER IF NOT.  THIS SUBROUTINE ASSUMES THAT ANY PART OF THE
C FLAME ABOVE THE HOT/COLD LAYER INTERFACE IS INVISIBLE TO THE
C TARGET UNLESS THE TARGET IS THE SAME AS THE SOURCE.  THE
C SUBROUTINE ALSO ASSUMES THAT THE TARGET AND SOURCE OBJECTS DO
C NOT OVERLAP UNLESS THE TARGET IS THE SAME AS THE SOURCE.
C
C EQUATIONS BY H. MITLER. ADAPTED FROM MARK5 VERSION BY J. GAHM,
C JULY 1981.
C
C INPUT PARAMETERS:
C     ISTATT -- THE STATE OF THE TARGET (ISTATT=5 -->
C               THE TARGET IS BURNING.)
C     KOF -- THE OBJECT NUMBER OF THE FIRE.
C     RFPF -- RADIUS OF THE FIRE AT THE PREVIOUS TIME STEP.
C     T -- FLAME TEMPERATURE.
C     TMOPF -- NEGATIVE OF THE MASS PYROLYSIS RATE OF THE
C              FIRE AT THE PREVIOUS TIME STEP.
C     HFPF -- HEIGHT OF THE FLAME OF THE FIRE AT THE PREVIOUS TIME STEP.
C     ZHOZZF -- HEIGHT OF THE BASE OF THE FIRE.
C     ZHOZZT -- HEIGHT OF THE TARGET.
C     ZHRZZ -- HEIGHT OF THE ROOM CONTAINING THE TWO OBJECTS.
C     ZRFZIF -- OBJECT RADIUS OF THE FIRE.
C     ZRFZIT -- OBJECT RADIUS OF THE TARGET.
C     ZUFZZ -- ABSORPTIVITY OF THE FLAME.
C     ZXOZZF -- X-COORDINTATE OF THE FIRE.
C     ZXOZZT -- X-COORDINATE OF THE TARGET.
C     ZYOZZF -- Y-COORDINATE OF THE FIRE.
```

```
C      ZYOZZT -- Y-COORDINATE OF THE TARGET.
C
C INPUT VARIABLES:
C      TMOZZF -- NEGATIVE OF MASS PYROLYSIS RATE OF THE FIRE.
C      ZHFZZF -- HEIGHT OF THE FLAMES OVER THE FIRE.
C      ZHLZZ -- DEPTH OF THE LAYER IN THE ROOM.
C      ZHPZZF -- HEIGHT OF THE PLUME OVER THE FIRE.
C      ZMOZZF -- MASS OF THE BURNING OBJECT.
C      ZRFZZF -- RADIUS OF THE FIRE.
C      ZRFZZT -- RADIUS OF THE FIRE ON THE TARGET OBJECT.
C
C OUTPUT VARIABLE:
C      FQ1P01 -- RADIATIVE FLUX FROM THE FIRE TO THE TARGET.
C
C NOTE THAT THE FOLLOWING INCLUDE FILE IS NEEDED ONLY TO GET THE
C MAXIMUM NUMBER OF OBJECTS IN THE RUN FOR THE DIMENSIONS OF VARIOUS
C ARRAYS.
$INCLUDE 9,DIMN.CMN
      PARAMETER (MO1 = MO*MR)
      PARAMETER (SIGMA = 5.67D-8)
      PARAMETER (PI = 3.1415927D0)
      PARAMETER (XMEAN = 0.7755D0)
C
      DIMENSION SET(MO1),ALPHAO(MO1)
      SAVE SET, ALPHAO, ISHELL
      LOGICAL SET
      DATA ISHELL,SET /0,MO1*.TRUE./
C
C IF FIRE IS BURNT OUT FLUX IS 0.
C
      IF (ZMOZZF .LE. 0.D0) THEN
        FQ1P01 = 0.D0
        GO TO 10
      END IF
C
C IF EITHER THE SOURCE OR THE TARGET IS IMMERSED IN THE LAYER,
C THE FLUX IS 0.
C
      IF (ZHOZZT.GE.ZHRZZ-ZHLZZ .OR. ZHFZZF.LE.1.D-10) THEN
        FQ1P01 = 0.D0
        GO TO 10
      END IF
C
      D = SQRT( (ZXOZZT-ZXOZZF)**2 + (ZYOZZT-ZYOZZF)**2 )
      F = SIGMA * T**4
C
C FIND THE HEIGHT OF THE FLAME
C
      IF (ZMOZZF .LT. .20D0) THEN
C
C REDUCE THE HEIGHT OF THE FLAME NEAR BURNOUT.
        IF (SET(KOF)) THEN
          ALPHAO(KOF) = HFPF/SQRT(ABS(TMOPF))
          SET(KOF) = .FALSE.
        END IF
        H = ALPHAO(KOF) * SQRT(ABS(TMOZZF))
      ELSE
        H = ZHFZZF
      END IF
C
C FIND TANGENT, SINE, AND COSINE OF SEMI-APEX ANGLE
C
      IF (1.04*H .GT. ZRFZZF) THEN
```

```
       TP = ZRFZZF/H
      ELSE
       TP = 1.D4
      END IF
      CO = 1.D0 / SQRT(1.D0+TP*TP)
      SI = TP * CO
C
      XO = ZHOZZT - ZHOZZF
C
C CASE: TARGET ABOVE THE FLAME TIP.
      IF (XO .GT. H) THEN
        FQ1PC1 = 0.D0
C
C CASE: FIRE BASE ABOVE TARGET.
      ELSE IF (XO .LT. 0.D0) THEN
C SLAB ASSUMED TO BE SQUARE.
        RZ = ZRFZIF * 0.886227D0
        R2 = ZRFZZT
        HZ = -XO
        DZ = D-RZ
C
C THE SECOND INEQUALITY IN THE "IF" STATEMENT BELOW APPLIES
C JUST SO LONG AS WE ASSUME THAT THE LAYER IS OPAQUE AS FAR
C AS POINTS EXTERNAL TO THE FLAME ARE CONCERNED.
        IF (H*(DZ+R2) .LT. HZ*RZ .OR. ZHPZZF*(DZ+R2) .LT. HZ*RZ) THEN
          FQ1PO1 = 0.D0
        ELSE
          RT = (DZ*DZ + HZ*HZ)**1.5D0
          IF (ZHPZZF .LT. H) THEN
            DV = ( H - ZHPZZF )**3
          ELSE
            DV = 0.D0
          END IF
          IF (D.NE.0.D0) BEAM = HZ * TP**2 * ((H*DZ-HZ*RZ)**3
     1             - DZ**3*DV) / (3.D0*D*D*DZ*RT)
          XL = .77D0 * (ZRFZZF - RZ*HZ*TP/DZ)
          FQ1PC1 = F * BEAM/XL * (1.D0-EXP(-ZUFZZ*XL))
          IF (ISTATT .EQ. 5) THEN
            TAU1 = ZUFZZ*ZRFZIT
            FQ1PC1 = FQ1PC1 * EXP(-TAU1)
          END IF
          IF (H*(DZ-R2) .LT. HZ*RZ) THEN
            CFE = (DZ - RZ*HZ/H) / R2
            FE = COS(CFE)
            FQ1PC1 = FQ1PC1 * (1.D0 + (CFE*SQRT(1.D0-CFE**2)-FE)/PI)
          END IF
        END IF
C
C CASE:  TARGET ABOVE THE FIRE.
      ELSE
        IF (D .LE. ZRFZZF) THEN
          X1 = H - XO
        ELSE
          X1 = MIN(ZHPZZF,H) - XO
        END IF
C
        IF (X1 .LE. 0.D0) THEN
          FQ1PC1 = 0.D0
          GO TO 10
        END IF
C
C CALCULATE RADIUS OF THE UPPER PART OF THE CONE
C
```

```
        TSI = D/(ZRFZZF - X0*TP)
C
C ED -- EFFECTIVE DISTANCE.
C
        IF (TSI .GE. 1.D0) THEN
          ED = D * (1.D0 + (.2787D0/(.4349D0 - TSI*TSI)))
        ELSE
          ED = D * .5068D0 * (TSI**2.8254D0)
        END IF
C
        XA = X0
        XB = (X0 + X1)
        ED2 = ED**2
C
        IF (X1 .LE. 100.0D0*X0) THEN
          FQ1P01 = X1 * XA * (1.0D0 / SQRT(ED2 + XA**2)
     1         - 1.0D0 / SQRT(ED2 + XA**2 + (ZRFZZF - XA*TP)**2))
        ELSE
          YA = XA/CO - ZRFZZF*SI
          YB = XB/CO - ZRFZZF*SI
          SA = ED2 + (ZRFZZF*CO)**2
          SB = SQRT(SA + YB**2)
          SA = SQRT(SA + YA**2)
          FQ1P01 = (SQRT(ED2 + XB**2) - SQRT(ED2 + XA**2)
     1                  - (SB-SA)*CO**2
     2                  + ZRFZZF*SI*CO**2*LOG((YA + SA)/(YB + SB)))
        ENDIF
C FQ1P01 HAS (PHI TC)*2/G.
        FQ1P01 = 2.0D0 * ZUFZZ * FQ1P01
C ARE YOU SURE IT IS NOT MULTIPIED BY 0.5?
C
C IF ED = 0 THEN THE TARGET IS THE SOURCE
C
        IF (ED .LE. 0.D0) THEN
          FQ1P01 = F * (1.D0 - EXP(-XMEAN*FQ1P01))
        ELSE IF (ISHELL .EQ. 0) THEN
          TAU = 1.27324D0*ZRFZZF*ZUFZZ*(1.D0+.84D0*(ZRFZZF**2)/(D*D))
          FQ1P01 = F * FQ1P01
          IF (TAU .GT. 1.0D-5) FQ1P01 = FQ1P01*(1.0D0 - EXP(-TAU))/TAU
          IF (ISTATT .EQ. 5) THEN
            TAU1 = ZUFZZ * ZRFZIT
            FQ1P01 =FQ1P01 * EXP(-TAU1)
          END IF
        END IF
      END IF
C
C LIMITS:
C
   10 CONTINUE
C     IF (FQ1P01 .LT. 0.D0) THEN
C       FQ1P01 = 0.D0
C     ELSE IF (FQ1P01 .GT. 2.D6) THEN
C       FQ1P01 = 2.D6
C     END IF
C
      RETURN
      END
CHECKED
      SUBROUTINE RNWOD2(ISTAT,ZHOZZ,ZHRZZ,ZKAZZ,ZLRZX,ZLRZY,
     1          ZUFZZ,ZXOZZ,ZYOZZ,ZHFZZ,ZHLZZ,ZKWZZ,ZRFZZ,ZULZZ,
     2          FQWOR1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
```

```
C CALCULATES THE RADIATIVE FLUX FROM THE WALLS
C AND CEILING IN A ROOM TO THE TOP SURFACE OF AN OBJECT. THE
C OBJECT MAY BE BURNING.
C
C EQUATIONS BY H. MITLER. ADAPTED FROM MARK5 VERSION BY J. GAHM,
C JULY 1981.
C
C INPUT PARAMETERS:
C      ISTAT -- THE STATE OF THE OBJECT (ISTAT=5 --> OBJECT
C               IS BURNING.)
C      ZHOZZ -- HEIGHT OF THE OBJECT.
C      ZHRZZ -- HEIGHT OF THE ROOM.
C      ZKAZZ -- AMBIENT TEMPERATURE OF THE ROOM.
C      ZLRZX -- LENGTH OF THE ROOM ALONG THE X-AXIS.
C      ZLRZY -- LENGTH OF THE ROOM ALONG THE Y-AXIS.
C      ZUFZZ -- ABSORPTIVITY OF THE FLAME.
C      ZXOZZ -- X-COORDINATE OF THE OBJECT.
C      ZYOZZ -- Y-COORDINATE OF THE OBJECT.
C
C INPUT VARIABLES:
C      ZHFZZ -- HEIGHT OF THE FLAMES OVER THE OBJECT.
C      ZHLZZ -- DEPTH OF THE LAYER IN THE ROOM.
C      ZKWZZ -- TEMPERATURE OF THE WALLS IN THE ROOM.
C      ZRFZZ -- RADIUS OF THE FIRE ON THE OBJECT.
C      ZULZZ -- ABSORPTIVITY OF THE HOT LAYER.
C
C OUTPUT VARIABLE:
C      FQWOR1 -- FLUX FROM THE WALLS TO THE SURFACE OF THE OBJECT.
C
       PARAMETER (PI = 3.14159270D0)
       PARAMETER (SIGMA = 5.67D-8)
       DIMENSION A(4),OPAC(4)
C
C 9/24/80. NEW GEOMETRIC APPROXIMATION TO THE VIEW FACTOR.
C
C FIND AREA OF THE 4 QUADRANTS.
       A(1) = (ZLRZX - ZXOZZ) * (ZLRZY - ZYOZZ)
       A(2) = (ZLRZX - ZXOZZ) * ZYOZZ
       A(3) = ZYOZZ * ZXOZZ
       A(4) = ZXOZZ * (ZLRZY - ZYOZZ)
C
C DL IS THE THICKNESS OF THE HOT LAYER BETWEEN THE OBJECT AND THE
C CEILING.
C
C DISTANCE FROM THE OBJECT TO THE BOTTOM OF THE HOT LAYER:
       XH = ZHRZZ - ZHOZZ - ZHLZZ
       XH2 = XH * XH
       IF (XH .GT. 0.D0) THEN
         DL = ZHLZZ
C FIND VIEW FACTOR IN GENERAL.
         VF = 0.D0
         DO 10 I=1,4
   10    VF = VF + A(I) / ( A(I) + .25D0 * PI * XH2 )
         VF = VF / 4.D0
       ELSE
         XH = 0.D0
         XH2 = 0.D0
         DL = ZHRZZ - ZHOZZ
         VF = 1.D0
       END IF
C
C NEXT FIND OPACITY THROUGH THE LAYER FOR EACH OF THE 4 QUADRANTS.
C
```

```
      XHDL2 = (XH + DL)**2
C
      DO 20 I=1,4
        R2 = 4.D0 * A(I) / PI
        IF (R2 .NE. 0.D0) OPAC(I) = 2.D0* ZULZZ * (1.D0+XHDL2/R2) *
     1          (DL+SQRT(R2+XH2)-SQRT(R2+XHDL2))
   20 CONTINUE
C
      ATTEN = 0.D0
      DO 30 I=1,4
   30 ATTEN = ATTEN + A(I)*EXP(-OPAC(I)) / (A(I)+.25D0*PI*XH2+1.D-10)
      ATTEN = ATTEN / 4.D0
      TW4 = ZKWZZ**4
      TA4 = ZKAZZ**4
      FQWOR1 = SIGMA * (ATTEN*TW4 + (1.D0-VF)*TA4)
C
C WHEN OBJECT IS BURNING FLUX IS REDUCED BY ABSORPTION
C IN THE FLAME.
C
      IF (ISTAT .EQ. 5) THEN
        TAU = ZRFZZ * ZUFZZ
        HYP = SQRT(ZRFZZ*ZRFZZ + ZHFZZ*ZHFZZ)
        IF (HYP .NE. 0.D0) THEN
         SI = ZRFZZ / HYP
        ELSE
         SI = 1.D0
        END IF
        FQWOR1 = FQWOR1 * ( EXP(-TAU) + SI ) / (1.D0 + SI)
      END IF
C
C LIMITS:
C
C     IF ( FQWOR1 .LT. 0.D0) THEN
C       FQWOR1 = 0.D0
C     ELSE IF ( FQWOR1 .GT. 2.D6) THEN
C       FQWOR1 = 2.D6
C     END IF
C
      RETURN
      END
      SUBROUTINE SPECFL(TMUZZ,ZYLDZ,ZYLWZ,ZYLSZ,ZYLMZ,ZYLOZ,
     1 TMDZF1,TMWZF1,TMSZF1,TMMZF1,TMOZF1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C THIS SUBROUTINE CALCULATES THE MASS FLOW OF EACH OF THE GASEOUS
C SPECIOUS THROUGH A VENT.
C
C INPUT PARAMETERS:
C
C INPUT VARIABLES:
C     TMUZZ -- THE (POSITIVE) MASS FLOW THROUGH A VENT.
C     ZYLDZ -- THE MASS CONCENTRATION OF CARBON DIOXIDE IN THE HOT
C              LAYER OF THE ROOM FROM WHICH THE FLOW ORIGINATES.
C     ZYLWZ -- THE MASS CONCENTRATION OF WATER IN THE HOT LAYER OF
C              THE ROOM FROM WHICH THE FLOW ORIGINATES.
C     ZYLSZ -- THE MASS CONCENTRATION OF SOOT IN THE SAME ROOM AS
C              ABOVE.
C     ZYLMZ -- THE MASS CONCENTRATION OF CARBON MONOXIDE IN THE
C              SAME ROOM AS ABOVE.
C     ZYLOZ -- THE MASS CONCENTRATION OF OXYGEN IN THE SAME ROOM AS
C              ABOVE.
C
C OUTPUT VARIABLES:
```

```
C      TMDZF1 -- THE MASS FLOW THROUGH THE VENT OF CARBON-DIOXIDE.
C      TMWZF1 -- THE MASS FLOW THROUGH THE VENT OF WATER VAPOR.
C      TMSZF1 -- THE MASS FLOW THROUGH THE VENT OF SOOT.
C      TMMZF1 -- THE MASS FLOW THROUGH THE VENT OF CARBON MONOXIDE.
C      TMOZF1 -- THE MASS FLOW THROUGH THE VENT OF OXYGEN.
C
       TMDZF1 = TMUZZ*ZYLDZ
       TMWZF1 = TMUZZ*ZYLWZ
       TMSZF1 = TMUZZ*ZYLSZ
       TMMZF1 = TMUZZ*ZYLMZ
       TMOZF1 = TMUZZ*ZYLOZ
       RETURN
       END
CHECKED
       SUBROUTINE SPREAD(A,T,ZRFZM,ZRFZO,PHI,ZRFZZ,TRFZZ1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE SPREAD RATE OF THE FIRE ON AN
C OBJECT.
C
C EQUATIONS BY H. MITLER.  ADAPTED FROM GFIRO3 CODE BY J. GAHM,
C JUNE 1981.
C
C INPUT PARAMETERS:
C      A -- FIRE SPREAD PARAMETER.
C      T -- FLAME TEMPERATURE.
C      ZRFZM -- MAXIMUM BURNING RADIUS OF THE OBJECT.
C      ZRFZO -- INITIAL BURNING RADIUS OF THE OBJECT.
C
C INPUT VARIABLES:
C      PHI -- FLUX INCIDENT ON THE SURFACE OF THE OBJECT.
C      ZRFZZ -- RADIUS OF THE FIRE ON THE OBJECT.
C
C OUTPUT VARIABLE:
C      TRFZZ1 -- SPREAD RATE OF THE FIRE ON THE OBJECT.
C
       PARAMETER (SIGMA = 5.67D-8)
C
       C = PHI / (SIGMA * T**4)
       IF (ZRFZZ .LE. 0.95D0*ZRFZM) THEN
         TRFZZ1 = A * C * (1.D0+C/2.D0+C*C/3.D0)
       ELSE
         TRFZZ1 = (ZRFZM-ZRFZZ) / 10.D0
       END IF
C
C LIMITS:
C
C      IF (ZRFZZ .GE. ZRFZM) THEN
C        TRFZZ1 = MIN(TRFZZ1,0.D0)
C      ELSE
C        IF (ZRFZZ .LE. ZRFZO) TRFZZ1 = MAX(TRFZZ1,0.D0)
C      END IF
C
       RETURN
       END
CHECKED
       SUBROUTINE SUTCNC(ZMSL,ZMLZZ,ZYLSZ1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE MASS CONCENTRATION OF SOOT
C IN THE HOT LAYER.
C
C EQS. BY H. EMMONS.  ADAPTED FROM LAYR04 CODE BY J. GAHM,
C JULY 1981.
C
C INPUT PARAMETERS:
C
C INPUT VARIABLES:
C      ZMSL -- MASS OF SOOT IN THE HOT LAYER.
C      ZMLZZ -- TOTAL MASS OF THE HOT LAYER.
C
C OUTPUT VARIABLE:
C      ZYLSZ1 -- MASS CONCENTRATION OF SOOT IN THE HOT LAYER.
C
       PARAMETER (YS = 0.D0)
C
       IF (ZMSL.LE.ZMLZZ .AND. ZMLZZ.NE.0.D0) THEN
         ZYLSZ1 = ZMSL / ZMLZZ
       ELSE
         ZYLSZ1 = YS
       END IF
C
C LIMITS:
C      IF (ZYLSZ1 .LT. YS ) THEN
C        ZYLSZ1 = YS
C      ELSE IF ( ZYLSZ1 .GT. 1.D0 ) THEN
C        ZYLSZ1 = 1.D0
C      END IF
C
       RETURN
       END
CHECKED
       SUBROUTINE SUTPRD(FS,TMOZZ,TMOS1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE RATE AT WHICH THE FIRE OF AN OBJECT
C PRODUCES SOOT.
C
C ADAPTED FROM LAYR04 CODE BY J. GAHM, AUGUST 1981.
C
C INPUT PARAMETERS:
C      FS -- MASS FRACTION OF SOOT EVOLVED IN THE BURNING
C            OF THE OBJECT.
C
C INPUT VARIABLES:
C      TMOZZ -- NEGATIVE OF MASS PYROLYSIS RATE OF THE OBJECT.
C
C OUTPUT VARIABLE:
C      TMOS1 -- MASS RATE OF EVOLUTION OF CO2 FROM THE OBJECT.
C
       TMOS1 = -FS * TMOZZ
C
       RETURN
       END
       SUBROUTINE TCO2M(TMODS,TMOSM,TMPSM,TMDZF,TMDL1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE RATE OF CHANGE OF MASS OF CARBON-
C DIOXIDE IN THE HOT LAYER.
C
C EQUATIONS BY H. EMMONS AND H. MITLER. ADAPTED FROM LAYR04
C CODE BY J. GAHM.
C
C INPUT PARAMETERS:
C
C INPUT VARIABLES:
```

```
C       TMODS -- RATE (KG/SEC) AT WHICH THE OBJECTS IN THE ROOM
C                PRODUCE CARBON-DIOXIDE.
C       TMOSM -- NEGATIVE OF THE SUM OF THE MASS PYROLOSIS RATES
C                OF ALL OBJECTS IN THE ROOM.
C       TMPSM -- RATE AT WHICH THE PLUMES OVER ALL THE OBJECTS IN
C                THE ROOM ADD MASS TO THE HOT LAYER.
C       TMDZF -- RATE AT WHICH CARBON DIOXIDE FLOWS OUT OF THE ROOM
C                THROUGH ALL THE VENTS IN THE ROOM.
C
C OUTPUT VARIABLE:
C       TMDL1 -- RATE OF INCREASE OF MASS OF CO2 IN THE HOT LAYER.
C
        TMDL1 = 5.D-4 * (TMPSM + TMOSM) + TMODS - TMDZF
        RETURN
        END
        SUBROUTINE TENPLM(ZKAZZ,ZKOPY,TEOZZ,TEPZR,TMOZZ,TMPZZ,TEPZZ1)
        IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE RATE AT WHICH THE PLUME OVER AN OBJECT
C ADDS ENERGY TO THE HOT LAYER.
C
C EQUATIONS BY H. EMMONS.  ADAPTED FROM PLUM CODE BY J. GAHM.
C
C INPUT PARAMETERS:
C       ZKAZZ -- AMBIENT TEMPERATURE IN THE ROOM CONTAINING THE PLUME.
C       ZKOPY -- PYROLOSIS TEMPERATURE OF THE OBJECT.
C
C INPUT VARIABLES:
C       TEOZZ -- RATE OF CHEMICAL ENERGY PRODUCTION IN THE COMBUSTION
C                OF THE OBJECT.
C       TEPZR -- HEAT LOSS RATE FROM THE FLAME OF THE OBJECT.
C       TMOZZ -- NEGATIVE OF MASS PYROLYSIS RATE OF THE OBJECT.
C       TMPZZ -- RATE AT WHICH THE PLUME OVER THE OBJECT ADDS MASS
C                TO THE HOT LAYER.
C OUTPUT VARIABLE:
C       TEPZZ1 -- RATE AT WHICH THE PLUME ADDS ENERGY TO THE HOT LAYER.
C
        PARAMETER ( CP = 1004.D0 )
C
C
C (2/3/79)  IN THE FOLLOWING EQN, WE REPLACED EQN (18) (TECHNICAL
C REPORT 34) WITH THE MORE PRECISE EQN (18') (FOOTNOTE ON P. 12).
C ALSO, THE SPECIFIC HEAT IN THE LAST ITEM SHOULD BE THAT OF THE
C PYROLYZED GAS RATHER THAN THAT OF AIR; HOWEVER, WE MUST USE CP
C (THAT OF AIR) BECAUSE WE HAVE USED CP IN THE EXPRESSION
C                 T(L) = E(L)/C(L)M(L)
C APPEARING IN SUBROUTINE LAYR, RATHER THAN THE CORRECT AVERAGE
C C(L).  WE ASSUME T(P) = 750. KELVIN FOR THE PRESENT.
        TEPZZ1 = - TEOZZ + ( TMPZZ + TMOZZ ) * CP * ZKAZZ
       1          -TEPZR - TMOZZ * CP * ZKOPY
C
C LIMITS:
C       IF ( TEPZZ1 .LT. 0.D0 ) THEN
C         TEPZZ1 = 0.D0
C       ELSE IF ( TEPZZ1 .GT. 2.D9 ) THEN
C         TEPZZ1 = 2.D9
C       END IF
C
        RETURN
        END
        SUBROUTINE TH2OM(TMOWS,TMWZF,TMWL1)
        IMPLICIT REAL*8 (A-H,O-Z)
C
```

```
C SUBROUTINE TO CALCULATE THE RATE OF INCREASE OF MASS OF
C WATER VAPOR IN THE HOT LAYER.
C
C EQUATIONS BY H. EMMONS AND H. MITLER.  ADAPTED FROM LAYRO4
C CODE BY J. GAHM, JULY 1981.
C
C INPUT PARAMETERS:
C
C INPUT VARIABLES:
C      TMOWS -- TOTAL RATE AT WHICH OBJECTS IN THE ROOM PRODUCE
C               H2O.
C      TMWZF -- RATE OF MASS OUTFLOW OF WATER VAPOR THROUGH ALL THE
C               VENTS IN THE ROOM.
C
C OUTPUT VARIABLE:
C      TMWL1 -- RATE OF INCREASE OF MASS OF WATER IN THE HOT LAYER.
C
       TMWL1 = TMOWS - TMWZF
       RETURN
       END
       SUBROUTINE TMNXM(TMOMS,TMMZF,TMML1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE RATE OF INCREASE OF MASS OF CARBON-
C MONOXIDE IN THE HOT LAYER.
C
C EQUATIONS BY H. EMMONS AND H. MITLER. ADAPTED FROM LAYRO4
C CODE BY J. GAHM.
C
C INPUT PARAMETERS:
C
C INPUT VARIABLES:
C      TMOMS -- TOTAL RATE (KG/SEC) AT WHICH CARBON-MONOXIDE IS
C               PRODUCED BY THE OBJECTS IN THE ROOM.
C      TMMZF -- MASS RATE OF OUTFLOW OF CARBON- MONOXIDE THROUGH ALL
C               THE VENTS IN THE ROOM.
C
C OUTPUT VARIABLE:
C      TMML1 -- RATE OF CHANGE OF MASS OF CARBON-MONOXIDE IN THE
C               HOT LAYER.
C
       TMML1 = TMOMS - TMMZF
       RETURN
       END
       SUBROUTINE TMPL(ZKAZZ,CP,ZELZZ,ZMLZZ,
     1                 ZKLZZ1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE TEMPERATURE OF THE HOT LAYER IN A ROOM
C
C EQUATIONS BY H. EMMONS AND H. MITLER. ADAPTED FROM LAYRO4 CODE BY
C J. GAHM.
C
C INPUT PARAMETERS:
C      ZKAZZ -- AMBIENT TEMPERATURE IN THE ROOM.
C      CP -- SPECIFIC HEAT OF AIR.
C
C INPUT VARIABLES:
C      ZELZZ -- ENERGY OF THE LAYER.
C      ZMLZZ -- MASS OF THE LAYER.
C
C OUTPUT VARIABLE:
C      ZKLZZ1 -- TEMPERATURE OF THE LAYER.
```

```
C
C
      IF (ZMLZZ .EQ. 0.D0) THEN
        ZKL = ZKAZZ
      ELSE
        ZKL = ZELZZ / ( CP * ZMLZZ )
      END IF
      ZKLZZ1 = ZKL
C
C LIMITS:
C     IF ( ZKLZZ1 .LT. ZKAZZ ) THEN
C       ZKLZZ1 = ZKAZZ
C     ELSE IF ( ZKLZZ1 .GT. 2500.D0 ) THEN
C       ZKLZZ1 = 2500.D0
C     END IF
C
      RETURN
      END
      SUBROUTINE TMPC2(DT,EB,KO,ZHOZZ,ZHRZZ,
     1     ZKAZZ,ZOAZZ,FQLOR,FQPOR,FQWOR,
     2     ZHLZZ,ZKLZZ,ZKOZZ,ZKOZZ1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALCULATES THE TEMPERATURE PROFILE WITHIN OBJECT KO USING A DISCRETE
C GRID. THE OBJECT IS HEATED ON THE UPPER SURFACE BY RADIATION AND THE
C HEAT DIFFUSES THROUGH IT BY CONDUCTION.  NOTE THAT THIS SUBROUITNE
C DIFFERS FROM THE OTHER PHYSICAL SUBROUITNES IN THAT THE TIME AND
C SPACE INTEGRATIONS NECESSARY TO SOLVE THE HEAT EQUATION ARE PERFORMED
C WITHIN THE SUBROUTINE ITSELF.  TO FACILITATE THIS CALCULATION THE
C SUBROUITNE MUST BE INITIALIZED AT THE BEGINNING OF EACH TIME STEP
C VIA A CALL TO THE ENTRY TMPC2I.
C
C EQUATIONS BY H. EMMONS
C CODED BY D. LAPP, N.BILLIKOPF, L. TRFETHEN
C AUGUST 1977
C RECODED BY M. SPIVAKOVSKY, AUGUST 1980.  EQS. BY H. MITLER.
C ADAPTED TO NEW PHYSICAL SUBROUTINE FORMAT BY J. GAHM.
C
C INPUT PARAMETERS:
C     DT -- SIZE OF THE TIME STEP.
C     EB -- EMMISIVITY/ABSORPTIVITY OF THE SURFACE OF THE OBJECT.
C     KO -- OBJECT NUMBER OF THE OBJECT.
C     ZHOZZ -- HEIGHT OF THE OBJECT.
C     ZHRZZ -- HEIGHT OF THE ROOM CONTAINING THE OBJECT.
C     ZKAZZ -- AMBIENT TEMPERATURE IN THE ROOM CONTAINING THE OBJECT.
C     ZOAZZ -- HEAT TRANSFER COEFFICIENT FOR OBJECTS.
C
C INPUT VARIABLES:
C     FQLOR -- RADIATIVE FLUX FROM THE HOT LAYER IN THE ROOM TO THE
C              SURFACE OF THE OBJECT.
C     FQPOR -- RADIATIVE FLUX FROM ALL THE PLUMES IN THE ROOM  TO THE
C              SURFACE OF THE OBJECT.
C     FQWOR -- RADIATIVE FLUX FROM THE WALLS IN THE ROOM TO THE SURFACE
C              OF THE OBJECT.
C     ZHLZZ -- DEPTH OF THE LAYER IN THE ROOM.
C     ZKLZZ -- TEMPERATURE OF THE HOT LAYER.
C     ZKOZZ -- SURFACE TEMPERATURE OF THE OBJECT.
C
C OUTPUT VARIABLE:
C     ZKOZZ1 -- SURFACE TEMPERATURE OF THE OBJECT.
C
C THE FOLLOWING INCLUDE FILE IS NEEDED ONLY TO GET THE MAXIMUM
C NUMBER OF OBJECTS FOR THE RUN AS A DIMENSION FOR ZKO.
```

```
$INCLUDE 9,DIMN.CMN
      PARAMETER (MO1 = MO*MR)
      PARAMETER ( MXLYRS = 20 )
      PARAMETER ( SIGMA = 5.67D-8 )
      PARAMETER ( ZOLZZ = 5.D0, ZOLZM = 50.D0 )
      DIMENSION FIRST(MO1),NOB(MO1),AO(MO1),BO(MO1),ZKO(MXLYRS,MO1)
      SAVE FIRST,NOB,AO,BO,ZKO
      LOGICAL FIRST
      DATA FIRST /MO1*.TRUE./
C
C FIND TOTAL FLUX IMPINGING ON THE SURFACE OF THE OBJECT
      FQOA = FQPOR+FQLOR+FQWOR
C
C DETERMINE IF THE OBJECT IS IN THE HOT LAYER
C
      XH = ZHRZZ - ZHLZZ - ZHOZZ
      IF (XH .GT. 0.0D0) THEN
        RHG = ZOAZZ
        TG = ZKAZZ
      ELSE
        RHG = MIN(ZOLZM, ZOLZZ + (ZOLZM-ZOLZZ)
     1            *(ZKLZZ-ZKAZZ)/100.D0)
        TG = ZKLZZ
      END IF
      FQO = EB*(FQOA-SIGMA * ZKOZZ**4 ) -
     1        RHG*(ZKOZZ-TG)
C
      TEMP = 1.D0 - AO(KO) * DT
      ZKOZZ1 = ZKO(1,KO)*TEMP +
     1            AO(KO)*DT*(ZKO(2,KO)+BO(KO)*FQO)
C
C LIMITS:
C     IF ( ZKOZZ1 .GT. 2000.D0 ) THEN
C       ZKOZZ1 = 2000.D0
C     ELSE IF ( ZKOZZ1 .LT. ZKAZZ ) THEN
C       ZKOZZ1 = ZKAZZ
C     END IF
C
      RETURN
C
C
      ENTRY TMPOZI(KO,ZGOZZ,DTMAX,ZNOZZ,ZJOZZ,ZKAZZ,DT,ZKOZZ)
C
C CALL THIS ENTRY ONCE FOR EACH OBJECT TO INITIALIZE AT THE
C BEGINNING OF EACH TIME-STEP.
C
C INPUT PARAMETERS:
C     KO -- OBJECT NUMBER OF THE OBJECT.
C     ZGOZZ -- THERMAL DIFFUSITY OF THE OBJECT.
C     DTMAX -- LARGEST ALLOWABLE SIZE FOR THE TIME STEP.
C     ZNOZZ -- THICKNESS OF THE OBJECT.
C     ZJOZZ -- THERMAL CONDUCTIVITY OF THE OBJECT.
C     ZKAZZ -- AMBIENT TEMPERATURE IN THE ROOM.
C     DT -- SIZE OF THE TIME STEP JUST COMPLETED.
C     ZKOZZ -- SURFACE TEMPERATURE OF THE OBJECT.
C
C
C IF THIS IS THE FIRST CALL OF THE RUN THEN DETERMINE HOW MANY LAYERS
C WILL BE USED IN THE DISCRETE GRID FOR EACH OBJECT.
      IF (FIRST(KO)) THEN
        DXM = SQRT(2.D0*ZGOZZ*DTMAX)
        M = INT(ZNOZZ/DXM) + 1
        NOB(KO) = MIN(M,MXLYRS)
        NOB(KO) = MAX(2,NOB(KO))
        DX=ZNOZZ/(NOB(KO)-1)
        AO(KO) = 2.D0*ZGOZZ/(DX*DX)
        BO(KO) = DX/ZJOZZ
        DO 30 I = 1,NOB(KO)
          ZKO(I,KO) = ZKAZZ
   30   CONTINUE
        FIRST(KO) = .FALSE.
      ELSE
        TEMP = 1.D0 - AO(KO)*DT
        TMP2 = ZKOZZ
        DO 40 I=2,NOB(KO)-1
          TMP1 = TMP2
          TMP2 = ZKO(I,KO)*TEMP + AO(KO)*(DT/2.D0)*
     1  (ZKO(I-1,KO)+ZKO(I+1,KO))
          ZKO(I-1,KO) = TMP1
   40   CONTINUE
        TMP1 = TMP2
        TMP2 = ZKO(NOB(KO),KO)*TEMP +
     1       ZKO(NOB(KO)-1,KO)*AO(KO)*DT
        ZKO(NOB(KO)-1,KO) = TMP1
        ZKO(NOB(KO),KO) = TMP2
      END IF
      RETURN
      END
      SUBROUTINE TMPW1(DT,JSIDE,KW,ZKAZZ,FQLWC,FQLWR,FQPWR,
     1      ZHLZZ,ZKWZZ,ZULZZ,ZKWZZ1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C CALCULATES THE TEMPERATURE PROFILE WITHIN WALL KW USING A DISCRETE
C GRID.  THE WALL IS HEATED OR COOLED ON EACH SIDE BY CONVECTION
C AND RADIATION AND THE HEAT DIFFUSES THROUGH IT BY CONDUCTION.
C NOTE THAT THIS SUBROUTINE DIFFERS FROM THE OTHER PHYSICAL SUB-
C ROUTINES IN THAT THE INTEGRATIONS NECESSARY TO SOLVE THE ONE
C DIMENSIONAL HEAT EQUATION ARE PERFORMED WITHIN THE PHYSICAL SUB-
C ROUTINE ITSELF.  TO FACILITATE THIS INTEGRATION, THE SUBROUTINE
C MUST BE INITIALIZED AT THE BEGINNING OF EACH TIME STEP VIA A CALL
C TO THE ENTRY TMPW1I FOR EACH WALL.
C
C EQUATIONS BY H. EMMONS
C CODED BY D. LAPP, N. BILLIKOPF, L. TREFETHEN, AUGUST 1977
C B. LONDON, JULY 1978
C M.SPIVAKOVSKY, AUGUST 1980
C ADAPTED TO THE NEW PHYSICAL SUBROUTINE FORMAT BY J. GAHM,
C JULY 1981.
C
C INPUT PARAMETERS:
C      DT -- SIZE OF THE CURRENT TIME STEP.
C      JSIDE -- INDEX TO DIFFERENTIATE BETWEEN THE OUTSIDE AND INSIDE OF
C               THE WALL.
C      KW -- WALL NUMBER.
C      ZKAZZ -- AMBIENT TEMPERATURE IN THE ROOM BORDERED BY THE JSIDE SID
C               OF THE WALL KW.
C
C INPUT VARIABLES:
C      FQLWC -- CONVECTIVE HEAT FLUX FROM THE HOT LAYER IN CONTACT WITH T
C               JSIDE SIDE OF THE WALL TO THE WALL.
C      FQLWR -- RADIATIVE FLUX FROM THE HOT LAYER IN CONTACT WITH THE JSI
C               SIDE OF THE WALL TO THE WALL.
C      FQPWR -- TOTAL RADIATIVE FLUX FROM ALL THE PLUMES IN THE ROOM
C               BORDERED BY THE JSIDE SIDE OF THE WALL TO THE WALL.
C      ZHLZZ -- DEPTH OF THE HOT LAYER IN THE ROOM.
C      ZKWZZ -- TEMPERATURE OF THE JSIDE SIDE OF THE WALL.
```

```
C      ZULZZ -- INFRA-RED ABSORPTIVITY OF THE HOT LAYER IN THE ROOM.
C
C OUTPUT VARIABLE:
C      ZKWZZ1 -- TEMPERATURE OF THE JSIDE SIDE OF THE WALL.
C
C THE FOLLOWING INCLUDE STATEMENT IS NEEDED ONLY TO PROVIDE THE
C MAXIMUM NUMBER OF WALLS IN THE RUN AS A DIMENSION FOR ZKW.
$INCLUDE 9,DIMN.CMN
      PARAMETER ( MXLYRS = 20 )
      PARAMETER ( SIGMA = 5.67D-8 )
      DIMENSION N(MW),AW(MW),BW(MW),FIRST(MW),ZKW(MXLYRS,MW)
      SAVE N,AW,BW,FIRST,ZKW
      LOGICAL FIRST
      DATA FIRST /MW*.TRUE./
C
C TOTAL FLUX IMPINGING ON THE WALL
      FQW=FQLWD+FQLWR+FQPWR
C
C NOW WE TAKE INTO ACCOUNT ENERGY
C LOSS BY RADIATION:
      FQW = FQW - SIGMA*(ZKWZZ**4 - EXP(-ZHLZZ*ZULZZ)*
     1           ZKAZZ**4)
C
C AN ADDITIONAL FACTOR OF .5 WAS REMOVED FROM THE LAST TERM OF
C THE EXPRESSION FOR FQW ON 9/25/80 BY J. GAHM.
      TEMP = 1.DO - AW(KW) * DT
      IF (JSIDE .EQ. 1) THEN
        NSLC = 1
        NXTSLC = NSLC + 1
      ELSE
        NSLC = N(KW)
        NXTSLC = NSLC - 1
      END IF
      ZKWZZ1=ZKW(NSLC,KW)*TEMP+
     1     AW(KW)*DT*(ZKW(NXTSLC,KW)+BW(KW)*FQW)
C
C LIMITS:
C      IF ( ZKWZZ1 .LT. ZKAZZ ) THEN
C        ZKWZZ1 = ZKAZZ
C      ELSE IF ( ZKWZZ1 .GT. 2500.DO ) THEN
C        ZKWZZ1 = 2500.DO
C      END IF
C
      RETURN
C
C
C
      ENTRY TMPW1I(KW,ZGWZZ,DTMAX,ZNWZZ,IWTTY,ZKAIN,ZKAOUT,ZJWZZ,
     1 ZKWIN,ZKWOUT,DT)
C
C THIS ENTRY SHOULD BE CALLED ONCE FOR EACH WALL AT THE BEGINNING OF
C EACH TIME STEP TO INITIALIZE THE DISCRETE GRID.
C
C INPUT PARAMETERS:
C      KW -- WALL NUMBER OF THE WALL.
C      ZGWZZ -- THERMAL DIFFUSIVITY OF THE WALL.
C      DTMAX -- SIZE OF THE LARGEST ALLOWABLE TIME STEP.
C      ZNWZZ -- THICKNESS OF THE WALL
C      IWTTY -- CHANNEL NUMBER FOR WRITES TO THE TERMINAL SCREEN.
C      ZKAIN -- AMBIENT TEMPERATURE ON THE INNER FACE OF THE WALL
C               (JSIDE = 1).
C      ZKAOUT -- AMBIENT TEMPERATURE ON THE OUTER FACE OF THE WALL
C                (JSIDE = 2).
```

```
C      ZJWZZ -- THERMAL CONDUCTIVITY OF THE WALL.
C      ZKWIN -- TEMPERATURE OF THE INSIDE SURFACE OF THE WALL.
C      ZKWOUT -- TEMPERATURE OF THE OUTSIDE SURFACE OF THE WALL.
C      DT -- SIZE OF THE TIME STEP JUST COMPLETED.
C
C AT THE BEGINNING OF THE RUN, DETERMINE HOW MANY SLICES TO USE
C IN THE DISCRETE GRID AND SET UP PARAMETERS FOR EACH WALL.
      IF (FIRST(KW)) THEN
        DXM=SQRT(2.DO*ZGWZZ*DTMAX)
        M=INT(ZNWZZ/DXM)+1
        N(KW)=MIN(M,MXLYRS)
        IF (N(KW) .LE. 1) N(KW)=2
        WRITE (IWTTY,20) KW,N(KW)
   20   FORMAT (' *** TMPW01: NO. OF GRID POINTS (WALL=',I2,') =',I3)
        DX=ZNWZZ/(N(KW)-1)
        AW(KW)=2.DO*ZGWZZ/(DX*DX)
        BW(KW)=DX/ZJWZZ
        DO 30 I=1,N(KW)
          ZKW(I,KW) = ZKAIN + DBLE(I-1)*(ZKAOUT-ZKAIN)/DBLE(N(KW)-1)
   30   CONTINUE
        FIRST(KW) = .FALSE.
      ELSE
        TEMP = 1.DO - AW(KW)*DT
        TMP2 = ZKWIN
        DO 40 I = 2,N(KW)-1
          TMP1 = TMP2
          TMP2 = ZKW(I,KW)*TEMP + AW(KW)*(DT/2.DO)*
     1  (ZKW(I-1,KW)+ZKW(I+1,KW))
          ZKW(I-1,KW) = TMP1
   40   CONTINUE
        ZKW(N(KW)-1,KW)=TMP2
        ZKW(N(KW),KW) = ZKWOUT
      END IF
      RETURN
      END
      SUBROUTINE TMSL(VMAZZ,ZHRZZ,ZLRZX,ZLRZY,TMPSM,TMUSM,ZMLZZ,
     1             TMLZZ1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUITNE TO CALCULATE THE RATE OF CHANGE OF MASS OF THE LAYER.
C
C EQUATIONS BY H. EMMONS AND H. MITLER.
C ADAPTED FROM LAYR04 CODE BY J. GAHM.
C
C INPUT PARAMETERS:
C      VMAZZ -- DENSITY OF AMBIENT AIR IN THE ROOM.
C      ZHRZZ -- HEIGHT OF THE ROOM.
C      ZLRZX -- LENGTH OF THE ROOM ALONG THE X-AXIS.
C      ZLRZY -- LENGTH OF THE ROOM ALONG THE Y-AXIS.
C
C INPUT VARIABLES:
C      TMPSM -- TOTAL MASS OUTFLOW FROM THE PLUMES IN THE ROOM
C               INTO THE HOT LAYER.
C      TMUSM -- TOTAL MASS OUTFLOW FROM THE HOT LAYER THROUGH
C               ALL THE VENTS IN THE ROOM.
C      ZMLZZ -- MASS OF THE HOT LAYER IN THE ROOM.
C
C OUTPUT VARIABLE:
C      TMLZZ1 -- RATE OF CHANGE OF MASS OF THE LAYER.
C
      TMLZZ1 = TMPSM - TMUSM
C
C LIMITS:
```

```
C      IF (TMLZZ1 .LT. -1000.D0) THEN
C        TMLZZ1 = -1000.D0
C      ELSE IF (TMLZZ1 .GT. 1000.D0) THEN
C        TMLZZ1 = 1000.D0
C      END IF
C      IF (ZMLZZ .LE. 0.D0) THEN
C        TMLZZ1 = MAX(TMLZZ1,0.D0)
C      ELSE IF (ZMLZZ .GE. VMAZZ*ZLRZX*ZLRZY*ZHRZZ) THEN
C        TMLZZ1 = MIN(TMLZZ1,0.D0)
C      END IF
C
      RETURN
      END
      SUBROUTINE TMSPLM(ALPHA,VMAZZ,ZKAZZ,TEOZZ,TMOZZ,ZHPZZ,
     1          ZRFZZ,TMPZZ1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE RATE AT WHICH THE PLUME OVER AN
C OBJECT ADDS MASS TO THE HOT LAYER.
C
C EQUATIONS BY H. EMMONS.  ADAPTED FROM PLUM CODE BY J. GAHM.
C
C INPUT PARAMETERS:
C      ALPHA -- PLUME ENTRAINMENT COEFFICIENT.
C      VMAZZ -- DENSITY OF AMBIENT AIR IN THE ROOM CONTAINING THE
C               PLUME.
C      ZKAZZ -- AMBIENT TEMPERATURE IN THE ROOM CONTAINING THE PLUME.
C
C INPUT VARIABLES:
C      TEOZZ -- NEGATIVE OF RATE OF ENERGY RELEASE IN THE COMBUSTION
C               OF THE OBJECT.
C      TMOZZ -- NEGATIVE OF MASS PYROLYSIS RATE OF THE OBJECT.
C      ZHPZZ -- HEIGHT OF THE PLUME OF THE OBJECT.
C      ZRFZZ -- RADIUS OF THE FIRE ON THE OBJECT.
C
C OUTPUT VARIABLE:
C      TMPZZ1 -- RATE AT WHICH THE PLUME OVER THE OBJECT ADDS MASS TO
C                THE HOT LAYER.
C
      PARAMETER ( PI = 3.14159270D0, G = 9.8D0, CP = 1004.D0 )
C
      IF ( ZHPZZ .LT. 0.D0 ) THEN
        TMPZZ1 = -TMOZZ
      ELSE
        X = ZRFZZ / ( 1.2D0 * ALPHA )
        HPX = ZHPZZ + X
        BP = 1.2D0 * ALPHA * HPX
        TMP = ( -25.D0 * G * TEOZZ ) /
     1           ( 48.D0 * ALPHA*ALPHA * PI * CP * ZKAZZ * VMAZZ )
C
C SAFETY AGAINST FLUCTUATIONS:
        IF ( TMP .LE. 0.D0 ) THEN
          TMPZZ1 = -TMOZZ
        ELSE
          U = ( TMP / HPX )**0.33333330D0
          UF = (TMP / X )**0.33333330D0
          TMPZZ1 = VMAZZ * PI * ( BP*BP * U - ZRFZZ*ZRFZZ * UF )
     1          - TMOZZ
        END IF
      END IF
C
C LIMITS:
C      IF ( TMPZZ1 .LT. 0.D0 ) THEN
```

```
C         TMPZZ1 = 0.D0
C       ELSE IF ( TMPZZ1 .GT. 1000.D0 ) THEN
C         TMPZZ1 = 1000.D0
C       END IF
C
        RETURN
        END
CHECKED
        SUBROUTINE TOXYM(BOXYS,TMOSM,TMPSM,TMOZF,TMOL1)
        IMPLICIT REAL*8 (A-H,O-Z)
C
C  SUBROUTINE TO CALCULATE THE RATE OF CHANGE OF MASS OF OXYGEN
C  IN THE HOT LAYER.
C
C  EQUATIONS BY H. EMMONS AND H. MITLER. ADAPTED FROM LAYRO4
C  CODE BY J. GAHM, AUGUST 1981.
C
C  INPUT PARAMETERS:
C
C  INPUT VARIABLES:
C       BOXYS -- RATE AT WHICH ALL THE OBJECTS IN THE ROOM BURN OXYGEN.
C       TMOSM -- NEGATIVE OF THE SUM OF THE MASS PYROLYSIS RATES
C                OF ALL OBJECTS IN THE ROOM.
C       TMPSM -- RATE AT WHICH THE PLUMES OVER ALL THE OBJECTS IN
C                THE ROOM ADD MASS TO THE HOT LAYER.
C       TMOZF -- MASS RATE OF OUTFLOW OF OXYGEN FROM THE HOT LAYER
C                THROUGH ALL THE VENTS IN THE ROOM.
C
C  OUTPUT VARIABLE:
C       TMOL1 -- RATE OF INCREASE OF MASS OF OXYGEN IN THE HOT LAYER.
C
        PARAMETER ( YO2 = .2318D0 )
C  YO2 -- MASS CONCENTRATION OF OXYGEN IN THE AMBIENT AIR.
        TMOL1 = YO2 * ( TMPSM + TMOSM ) - BOXYS - TMOZF
        RETURN
        END
        SUBROUTINE TSUTM(TMOSS,TMSZF,TMSL1)
        IMPLICIT REAL*8 (A-H,O-Z)
C
C  SUBROUTINE TO CALCULATE THE RATE OF CHANGE OF MASS OF SOOT
C  IN THE HOT LAYER.
C
C  EQUATIONS BY H. EMMONS AND H. MITLER.  ADAPTED FROM LAYRO4
C  CODE BY J. GAHM, JULY 1981.
C
C  INPUT PARAMETERS:
C
C  INPUT VARIABLES:
C       TMOSS -- TOTAL RATE AT WHICH ALL OBJECTS IN THE ROOM
C                PRODUCE SOOT.
C       TMSZF -- MASS RATE OF OUTFLOW OF SOOT FROM THE HOT LAYER
C                THROUGH ALL THE VENTS IN THE ROOM.
C
C  OUTPUT VARIABLE:
C       TMSL1 -- RATE OF INCREASE OF MASS OF SOOT IN THE HOT LAYER.
C
        TMSL1 = TMOSS - TMSZF
        RETURN
        END
        SUBROUTINE VNTCVR(ZBVZZ,ZHTZZ,ZHVZZ,ZHLZZ,AVL1)
        IMPLICIT REAL*8 (A-H,O-Z)
C
C  SUBROUTINE TO CALCULATE THE AREA OF A VENT COVERED BY THE HOT
```

```
C LAYER.
C
C EQUATIONS BY H. MITLER. ADAPTED FROM RDNL CODE BY J. GAHM.
C
C INPUT PARAMETERS:
C      ZBVZZ -- WIDTH OF THE VENT.
C      ZHTZZ -- DISTANCE FROM THE CEILING TO THE TRANSOM
C               OF THE VENT.
C      ZHVZZ -- HEIGHT OF THE VENT.
C
C INPUT VARIABLES:
C      ZHLZZ -- DEPTH OF THE LAYER IN THE ROOM CONTAINING THE VENT.
C
C OUTPUT VARIABLE:
C      AVL1 -- AREA OF THE VENT COVERED BY THE HOT LAYER.
C
       HV = ZHLZZ - ZHTZZ
       IF ( HV .LT. 0.D0 ) THEN
         HV = 0.D0
       ELSE IF (HV .GT. ZHVZZ ) THEN
         HV = ZHVZZ
       END IF
       AVL1 = HV * ZBVZZ
       RETURN
       END
       SUBROUTINE WTRCNC(ZMWL,ZMLZZ,ZYLWZ1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE TO CALCULATE THE MASS CONCENTRATION OF WATER
C VAPOR IN THE HOT LAYER.
C
C EQS. BY H. EMMONS.  ADAPTED FROM LAYR04 CODE BY J. GAHM,
C JULY 1981.
C
C INPUT PARAMETERS:
C
C INPUT VARIABLES:
C      ZMWL -- MASS OF WATER VAPOR IN THE HOT LAYER.
C      ZMLZZ -- TOTAL MASS OF THE HOT LAYER.
C
C OUTPUT VARIABLE:
C      ZYLWZ1 -- MASS CONCENTRATION OF WATER VAPOR
C                IN THE HOT LAYER.
C
       PARAMETER (YW = 0.D0)
       IF (ZMWL.LE.ZMLZZ .AND. ZMLZZ .NE. 0.D0) THEN
         ZYLWZ1 = ZMWL / ZMLZZ
       ELSE
         ZYLWZ1 = YW
       END IF
C
C LIMITS:
C      IF (ZYLWZ1 .LT. YW ) THEN
C        ZYLWZ1 = YW
C      ELSE IF ( ZYLWZ1 .GT. 1.D0 ) THEN
C        ZYLWZ1 = 1.D0
C      END IF
C
       RETURN
       END
       SUBROUTINE WTRPRD(FH2O,TMOZZ,TMOW1)
       IMPLICIT REAL*8 (A-H,O-Z)
C
```

```
C SUBROUTINE TO CALCULATE THE RATE AT WHICH THE FIRE OF AN OBJECT
C PRODUCES WATER VAPOR.
C
C ADAPTED FROM LAYR04 CODE BY J. GAHM, AUGUST 1981.
C
C INPUT PARAMETERS:
C     FH2O -- MASS FRACTION OF WATER VAPOR EVOLVED IN THE BURNING
C             OF THE OBJECT.
C
C INPUT VARIABLES:
C     TMOZZ -- NEGATIVE OF MASS PYROLYSIS RATE OF THE OBJECT.
C
C OUTPUT VARIABLE:
C     TMCW1 -- MASS RATE OF EVOLUTION OF CO2 FROM THE OBJECT.
C
      TMCW1 = -FH2O * TMOZZ
C
      RETURN
      END
C
C
C        SUBROUTINE EMSVTY
C
C
C
C        SUBROUTINE EMSVTY COMPUTES THE EMISSIVITIES OF ISOTHERMAL AND
C        HOMOGENEOUS MIXTURES OF SOOT, CO2 AND H2O AT A TOTAL PRESSURE O
C        1 ATMOSPHERE. THE ACCURACY OF SUBROUTINE EMSVTY IS BETTER THAN
C        5 PERCENT AS COMPARED WITH SPECTRAL CALCULATIONS AND EXPERIMENT
C        MEASUREMENTS.
C
C
C        INPUTS:
C
C
C        SUBROUTINE EMSVTY REQUIRES 5 INPUT PARAMETERS
C        1.  ZKLED (IN INVERSE METRES):  THE ABSORPTION COEFFICIENT OF S
C        A WAVELENGTH OF 0.94 MICROMETRES. ZKLED IS (APPROXIMATELY) RELA
C        TO THE SOOT VOLUME FRACTION, FV, BY THE EQUATION ,
C        ZKLED = 7FV/0.94E-6. ZKLED MUST BE GREATER THAN OR EQUAL TO ZER
C
C        2.  PATHL (IN METRES):  MIXTURE PATHLENGTH. PATHL MUST BE GREAT
C        THAN OR EQUAL TO ZERO.
C
C        3.  T (IN DEGREES KELVIN):  GAS TEMPERATURE. T MUST BE GREATER
C        THAN OR EQUAL TO 300.00, AND LESS THAN OR EQUAL TO 2000.00.
C
C        4.  PCO2 (IN ATMOSPHERES):  PARTIAL PRESSURE OF CO2 IN A MIXTUR
C        AT A TOTAL PRESSURE OF 1 ATMOSPHERE.  PCO2 MUST BE GREATER THAN
C        EQUAL TO ZERO, AND LESS THAN OR EQUAL TO 1.0.  FOR PCO2 LESS TH
C        0.0011, THE CONTRIBUTION OF CO2 TO THE MIXTURE EMISSIVITY IS AS
C        TO BE ZERO.  FOR (PCO2*PATHL) LESS THAN 0.0011 ATM-METRE, THE
C        CONTRIBUTION OF CO2 TO THE MIXTURE EMISSIVITY IS ASSUMED TO BE
C        IF (PCO2*PATHL) EXCEEDS 5.98 ATM-METRE, SUBROUTINE EMSVTY DOES
C        CALCULATE THE CONTRIBUTION OF CO2 TO THE MIXTURE EMISSIVITY.  F
C        CASES, THE CONTRIBUTION OF CO2 TO THE MIXTURE EMISSIVITY IS SET
C        TO ZERO.
C
C        5.  PH2O (IN ATMOSPHERES):  PARTIAL PRESSURE OF H2O IN A MIXTUR
C        A TOTAL PRESSURE OF 1 ATMOSPHERE.  PH2O MUST BE GREATER THAN OR
C        TO ZERO, AND LESS THAN OR EQUAL TO 1.0.  (PCO2+PH2O) MUST BE LE
C        THAN OR EQUAL TO 1.0. FOR PH2O LESS THAN 0.0011, THE CONTRIBUTI
C        OF H2O TO THE MIXTURE EMISSIVITY IS ASSUMED TO BE ZERO.
```

```
C          FOR (PH2O*PATHL) LESS THAN 0.0011 ATM-METRE, THE CONTRIBUTION O
C          TO THE MIXTURE EMISSIVITY IS ASSMUED TO BE ZERO.  IF (PH2O*PATH
C          EXCEEDS 5.98 ATM-METRE, SUBROUTINE EMSVTY DOES NOT CALCULATE TH
C          CONTRIBUTION OF H2O TO THE MIXTURE EMISSIVITY. FOR SUCH CASES,
C          CONTRIBUTION OF H2O TO THE MIXTURE EMISSIVITY IS SET EQUAL TO Z
C
C
C          OUTPUT:
C
C
C          SUBROUTINE EMSVTY RETURNS EMISS - THE (DIMENSIONLESS) EMISSIVIT
C          A MIXTURE OF SOOT, CO2 AND H2O IN THE TEMPERATURE RANGE 300 TO
C          DEGREES KELVIN; A GAS PARTIAL PRESSURE RANGE 0.0 TO 1.0 ATMOSPH
C          AND A PRESSURE-PATHLENGTH RANGE 0.0 TO 5.98 ATM-METRES.
C
C
C          THE FOLLOWING SUBROUTINES MUST BE USED IN CONJUNCTION WITH EMSV
C          SOOT, PENTA, ASYMP, EGAS, DLECK, SCRTCH AND CHEBY.
C
C
C          QUESTIONS REGARDING EMSVTY MAY BE ADDRESSED TO
C          ASHOK T. MODAK
C          FACTORY MUTUAL RESEARCH CORPORATION
C          NORWOOD, MASSACHUSETTS  02062  USA
C          TELEPHONE (617)762-4300 EXT546
C
C
C          COMPUTATION TIME:
C
C
C          EACH CALL ON SUBROUTINE EMSVTY TAKES 11.85 MILLISECONDS
C          OF CPU TIME ON AN IBM 370/158.
C
C
C
C
C
C
C
C
C
C
      SUBROUTINE EMSVTY(ZKLED,PATHL,T,PCO2,PH2O,EMISS)
      IMPLICIT REAL*8 (A-H,O-Z)
      IF(ZKLED.LE.0.0D0)  ES=0.0D0
C        ES - SOOT EMISSIVITY
      IF(ZKLED.LE.0.0D0)  GOTO 10
      CALL SOOT(ZKLED,PATHL,T      ,TAUS)
C        TAUS - SOOT TRANSMISSIVITY
      ES=1.D0-TAUS
   10 CONTINUE
      IF(PCO2.LE.0.0D0      .AND.    PH2O.LE.0.0D0)  EG=0.0D0
C        EG - EMISSIVITY OF GASEOUS SPECIES:CO2 AND H2O
      IF(PCO2.LE.0.0D0      .AND.    PH2O.LE.0.0D0)  GOTO 20
      EG=EGAS(PATHL,PCO2,PH2O,T)
   20 CONTINUE
      EMISS=ES+EG-ES*EG
      RETURN
      END
C
C
C
C
```

```
C
C
      SUBROUTINE SOOT(ZKLED,PATHL,TBLACK,TAUS)
      IMPLICIT REAL*8 (A-H,O-Z)
C         SUBROUTINE SOOT COMPUTES THE TRANSMISSIVITY (TAUS) OF PATH (PAT
C         OF SOOT TO A BLACKBODY RADIATION SOURCE AT A GIVEN
C         TEMPERATURE (TBLACK).
C
      IF(ZKLED.LE.  0.D0)  GOTO 10
      ARG=1.D0+ZKLED*PATHL*TBLACK*6.5337D-5
C         TBLACK - SOURCE TEMPERATURE OR GAS TEMPERATURE
      CALL PENTA(ARG,V)
C         SUBROUTINE PENTA COMPUTES THE PENTAGAMMA FUNCTION
C         ARG - ARGUMENT  OF THE PENTAGAMMA FUNCTION
      TAUS=V*.1539897336D0
C         TAUS - SOOT TRANSMISSIVITY
      RETURN
   10 TAUS=1.D0
      RETURN
      END
C
C
C
C
C
C
C
      SUBROUTINE PENTA(X,V)
      IMPLICIT REAL*8 (A-H,O-Z)
C         SUBROUTINE PENTA RETURNS THE VALUE V OF THE PENTAGAMMA FUNCTION
C         OF ARGUMENT X.  RECURRENCE FORMULA 6.4.6, AND ASYMPTOTIC FORMUL
C         6.4.14 (PAGE 260) OF ABRAMOWITZ AND STEGUN ARE USED IN
C         THIS CALCULATION.
C
C
      IF(X.GE.4.D0)  GOTO 30
      IF(X.GE.3.D0)  GOTO 20
      IF(X.GE.2.D0)  GOTO 10
      S=(1.D0/(X+2.D0)**4+1.D0/(X+1.D0)**4+1.D0/X**4)*6.D0
      Z=X+3.D0
      CALL ASYMP(Z,ZV)
      GOTO 40
   10 CONTINUE
      S=(1.D0/(X+1.D0)**4+1.D0/X**4)*6.D0
      Z=X+2.D0
      CALL ASYMP(Z,ZV)
      GOTO 40
   20 CONTINUE
      S=6.D0/X**4
      Z=X+1.D0
      CALL ASYMP(Z,ZV)
      GOTO 40
   30 CONTINUE
      S=0.D0
      CALL ASYMP(X,ZV)
   40 CONTINUE
      V=ZV+S
      RETURN
      END
C
C
C
      SUBROUTINE ASYMP(Z,ZV)
      IMPLICIT REAL*8 (A-H,O-Z)
C         SUBROUTINE ASYMP COMPUTES THE ASYMPTOTIC EXPANSION FOR THE
C         PENTAGAMMA FUNCTION.
C
      ZI1=1.D0/Z
      ZI2=ZI1*ZI1
      ZI3=ZI1*ZI2
      ZV=ZI3*((2.D0+3.D0*ZI1)+ZI2*(2.D0+ZI2*(-1.D0+ZI2*
     1   (1.333333333D0+ZI2*(-3.D0+10.D0*ZI2)))))
      RETURN
      END
C
C
C
C
C
      FUNCTION EGAS(PATHL,PC,PW,T)
      IMPLICIT REAL*8 (A-H,O-Z)
C         FUNCTION EGAS COMPUTES THE EMISSIVITY OF A GIVEN PATH (PATHL)
C         OF A MIXTURE OF CO2 AND H2O AT TEMPERATURE T.
C
C         PC - PARTIAL PRESSURE OF CO2
C         PW - PARTIAL PRESSURE OF H2O
      EGAS=0.0D0
      IF(T.LT.300.D0  .OR.  T.GT.2000.D0) RETURN
      EC=0.0D0
      IF(PC.LT.0.0011D0  .OR.  PC.GT.1.0D0)  GOTO 10
      PCL=PC*PATHL
      IF(PCL.LT.0.0011D0  .OR.  PCL.GT.5.98D0)  GOTO 10
      CALL SCRTCH(PC,PCL,T,1,EC)
   10 CONTINUE
      IF(PW.LT.0.0011D0  .OR.  PW.GT.1.0D0)  GOTO 20
      PWL=PW*PATHL
      IF(PWL.LT.0.0011D0  .OR.  PWL.GT.5.98D0)  GOTO 20
      CALL SCRTCH(PW,PWL,T,2,EW)
      EGAS=EC+EW
      IF(EC.LE.0.0D0)  RETURN
      PCPW=PC+PW
      XI=PW/PCPW
      IF(XI.LT.0.01D0)  RETURN
      PCWL=PCPW*PATHL
      IF(PCWL.LT.0.1D0   )  RETURN
      DELE=DLECK(XI,PCWL,T)
      EGAS=EGAS-DELE
      RETURN
   20 CONTINUE
      EGAS=EC
      RETURN
      END
C
C
C
      FUNCTION DLECK(X,PL,T)
      IMPLICIT REAL*8 (A-H,O-Z)
C         SUBROUTINE DLECK COMPUTES THE 2.7 AND 15 MICROMETER OVERLAP
C         CORRECTION FOR MIXTURES OF CO2 AND H2O.  THE OVERLAP CORRECTION
C         IS COMPUTED BY USING A TEMPERATURE-ADJUSTED VERSION OF THE
C         OVERLAP CORRECTION FACTOR SUGGESTED BY B. LECKNER
C         (COMBUSTION AND FLAME VOLUME 19 PAGES 33-48, 1972)
      IF(PL.LT.0.1D0)   GOTO 10
      TERM=X/(10.7D0+101.D0*X)  -X**10.4D0/111.7D0
      TERM2=LOG10(101.325D0*PL)
      TERM2=TERM2**2.76D0
      TT=T/1000.D0
      TT2=TT*TT
      A=-1.0204082D0
      B=2.2448979D0
      C=-0.23469386D0
      TERM3=A*TT2+  B*TT  +  C
C        TERM3 REPRESENTS THE TEMPERATURE ADJUSTMENT
      DLECK=TERM *TERM2*TERM3
      RETURN
   10 DLECK=0.0D0
      RETURN
      END
C
C
C
      SUBROUTINE SCRTCH(P,PL,T,INDEX,V)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION CC(3,4,4),CW(3,4,4),         SC(3,4,4)
      IF(INDEX.EQ.2)  GOTO 10
C        CC REPRESENTS AN ARRAY OF 48 COEFFICIENTS FOR CO2
C
C
C
      CC(1,1,1)=-0.2754568D+01
      CC(1,1,2)=-0.2997857D+00
      CC(1,1,3)=-0.1232494D+00
      CC(1,1,4)=0.1279287D-01
      CC(1,2,1)=0.1503051D+01
      CC(1,2,2)=0.3156449D+00
      CC(1,2,3)=0.1058126D-01
      CC(1,2,4)=-0.3729625D-01
      CC(1,3,1)=-0.2474119D+00
      CC(1,3,2)=-0.3323846D-01
      CC(1,3,3)=-0.1819471D-01
      CC(1,3,4)=0.2289789D-01
      CC(1,4,1)=0.4994029D-01
      CC(1,4,2)=-0.1986786D-02
      CC(1,4,3)=0.3007898D-02
      CC(1,4,4)=-0.1175598D-02
      CC(2,1,1)=0.5737722D-02
      CC(2,1,2)=-0.9328458D-02
      CC(2,1,3)=0.2906266D-02
      CC(2,1,4)=0.4227520D-03
      CC(2,2,1)=-0.3151784D-02
      CC(2,2,2)=0.5632821D-02
      CC(2,2,3)=-0.3260295D-02
      CC(2,2,4)=0.7065884D-03
      CC(2,3,1)=0.1668751D-03
      CC(2,3,2)=-0.7326533D-03
      CC(2,3,3)=0.3639855D-03
      CC(2,3,4)=0.3228318D-03
      CC(2,4,1)=0.7386638D-03
      CC(2,4,2)=-0.7277073D-03
      CC(2,4,3)=0.5925968D-03
      CC(2,4,4)=-0.2021413D-03
      CC(3,1,1)=0.3385611D-02
      CC(3,1,2)=-0.5439185D-02
      CC(3,1,3)=0.1764560D-02
      CC(3,1,4)=0.3035031D-03
      CC(3,2,1)=-0.1862700D-02
      CC(3,2,2)=0.3236275D-02
      CC(3,2,3)=-0.1952250D-02
      CC(3,2,4)=0.3474022D-03
      CC(3,3,1)=0.1204807D-03
      CC(3,3,2)=-0.4479927D-03
      CC(3,3,3)=0.2497521D-03
      CC(3,3,4)=0.1812996D-03
      CC(3,4,1)=0.4218169D-03
      CC(3,4,2)=-0.4046608D-03
      CC(3,4,3)=0.3256861D-03
      CC(3,4,4)=-0.9514981D-04
      GOTO20
   10 CONTINUE
C        CW REPRESENTS AN ARRAY OF 48 COEFFICIENTS FOR H2O
C
C
      CW(1,1,1)=-0.2594279D+01
      CW(1,1,2)=-0.7118472D+00
      CW(1,1,3)=-0.9956839D-03
      CW(1,1,4)=0.1226560D-01
      CW(1,2,1)=0.2510331D+01
      CW(1,2,2)=0.6481808D+00
      CW(1,2,3)=-0.3330597D-01
      CW(1,2,4)=-0.5524345D-02
      CW(1,3,1)=-0.4191636D+00
      CW(1,3,2)=-0.1375180D+00
      CW(1,3,3)=0.3877930D-01
      CW(1,3,4)=0.8862328D-03
      CW(1,4,1)=-0.3223912D-01
      CW(1,4,2)=-0.1820241D-01
      CW(1,4,3)=-0.2223143D-01
      CW(1,4,4)=-0.5940781D-03
      CW(2,1,1)=0.1126869D+00
      CW(2,1,2)=-0.8133829D-01
      CW(2,1,3)=0.1514940D-01
      CW(2,1,4)=0.1393880D-02
      CW(2,2,1)=-0.9298805D-02
      CW(2,2,2)=0.4550660D-01
      CW(2,2,3)=-0.2082008D-01
      CW(2,2,4)=0.2013361D-02
      CW(2,3,1)=-0.4375032D-01
      CW(2,3,2)=0.1924597D-01
      CW(2,3,3)=0.8859877D-02
      CW(2,3,4)=-0.4618414D-02
      CW(2,4,1)=0.7077876D-02
      CW(2,4,2)=-0.2096188D-01
      CW(2,4,3)=0.1458262D-02
      CW(2,4,4)=0.3851421D-02
      CW(3,1,1)=0.5341517D-01
      CW(3,1,2)=-0.3407693D-01
      CW(3,1,3)=0.4354611D-02
      CW(3,1,4)=0.1492038D-02
      CW(3,2,1)=-0.4708178D-02
      CW(3,2,2)=0.2086886D-01
      CW(3,2,3)=-0.9477533D-02
      CW(3,2,4)=0.6153272D-03
      CW(3,3,1)=-0.2104622D-01
      CW(3,3,2)=0.7515796D-02
      CW(3,3,3)=0.5965509D-02
      CW(3,3,4)=-0.2756144D-02
      CW(3,4,1)=0.4318975D-02
      CW(3,4,2)=-0.1005744D-01
      CW(3,4,3)=0.4091084D-03
      CW(3,4,4)=0.2550435D-02
   20 CONTINUE
      X=LOG(P)/3.450D+1.0D0
      Y=(LOG(PL)+2.555D0)/4.345D0
      Z=(T-1150.D0)/ 850.D0
      V=0.0D0
      DO 50 II=1,3
        I=II-1
        CALL CHEBY(I,X,TIX)
        V6=0.0D0
        DO 40 JJ=1,4
          J=JJ-1
          CALL CHEBY(J,Y,TJY)
          V7=0.0D0
          DO 30 KK=1,4
            K=KK-1
            CALL CHEBY(K,Z,TKZ)
            IF(INDEX.EQ.1)SC(II,JJ,KK)=CC(II,JJ,KK)
            IF(INDEX.EQ.2)SC(II,JJ,KK)=CW(II,JJ,KK)
            V7=V7+TKZ*SC(II,JJ,KK)
   30     CONTINUE
          V6=V6+V7*TJY
   40   CONTINUE
        V=V+V6*TIX
   50 CONTINUE
      V=EXP(V)
      RETURN
      END
C
C
C
C
C
      SUBROUTINE CHEBY(N,X,V)
      IMPLICIT REAL*8 (A-H,O-Z)
C        V REPRESENTS VALUE OF CHEBYSHEV POLYNOMIAL OF ORDER N
C        AND ARGUMENT X.
      V=1.D0
      IF(N)  10,10,20
   10 RETURN
C
   20 V=X
      IF(N-1)  10,10,30
   30 F=X+X
      VM1=X
      VM2=1.D0
      DO 40 I=2,N
        V=F*VM1-VM2
        VM2=VM1
        VM1=V
   40 CONTINUE
      RETURN
      END
      SUBROUTINE RTIME(IFLAG)
C
C THIS SUBROUTINE RETURNS TO THE USER DIFFERENCES IN CPU TIME,
C FROM WHEN FIRST CALLED WITH IFLAG = 0, TO WHEN LATER CALLED
C WITH IFLAG = 1.
C
C     CALL CPUTIM(ITIME)
      GO TO (10,20) IFLAG+1
   10 ITIME0 = ITIME
      RETURN
C
   20 IDIFFT = ITIME - ITIME0
      DSEC = FLOAT(IDIFFT) / 100.
      MIN = DSEC / 60.
      SEC = AMOD(DSEC,60.)
      WRITE (6,30) MIN, SEC
      WRITE (19,30) MIN, SEC
   30 FORMAT (/,' ELAPSED CPU TIME: ',I3,' MINUTES, ',F6.3,' SECONDS')
      RETURN
      END
```

PART 3 - COMMON BLOCKS

```
C FILE VAR.CMN
C
C THIS COMMON BLOCK CONTAINS THE ARRAYS WHICH STORE THE OUTPUT OF
C THE CALLS TO THE VARIOUS PHYSICAL SUBROUTINES IN UNPACKED FORM.  THE
C COMMON BLOCK IS DIVIDED INTO THREE SECTIONS AS FOLLOWS:
C      SECTION 1 IS ROUGHLY EQUIVALENT TO THE MARKS VERSION OF COMMON
C /VAR/.  THE ARRAY NAMES IN THIS SECTION CORRESPOND TO THE NAMES OF
C THE VARIABLES FOR WHICH THE NUMERICAL MACHINERY IS ACTUALLY TRYING
C TO SOLVE.  RECALL THAT THE NUMERICAL PACKAGE SEES A PROBLEM IN THE
C FORM
C                    F(W,X,Y,t) = 0.
C                    G(W,X,Y,t) = X'
C                    H(W,X,Y,t) = Y
C                    L(W,X,Y,t) = Z.
C WHEN THE PROBLEM IS REPRESENTED IN THIS FORM, WE CAN SAY THAT THE
C FIRST SECTION OF /VAR/ CONTAINS A NAME CORRESPONDING TO EACH COM-
C PONENT OF EACH OF THE VECTORS W, X, Y, AND Z.
C      SECTION 2 OF /VAR/ IS USED TO STORE THE OUTPUT OF THE ROOT-
C FINDING EQUATIONS.  THUS, THE NAMES IN THIS SECTION OF /VAR/
C CORRESPOND TO THE COMPONENTS OF THE VECTOR VALUED FUNCTION F.
C      SECTION 3 OF /VAR/ IS USED TO STORE DERIVATIVE INFORMATION.
C THUS, THIS SECTION IS USED TO STORE THE OUTPUT OF THE VECTOR
C VALUED FUNCTION G.  THE SET OF NAMES IN SECTION 3 OF /VAR/
C CORRESPONDS TO A SUBSET OF THE NAMES IN SECTION 1 EXCEPT
C THAT EACH ARRAY NAME IN SECTION 3 IS PREFIXED BY THE LETTER D TO
C INDICATE THAT THE ARRAY IS USED TO STORE THE DERIVATIVE OF THE
C VARIABLE WITH CORRESPONDING NAME IN SECTION 1.  THUS, DZMOZZ IS
C USED TO STORE THE DERIVATIVE OF ZMOZZ AND DAQWZZ IS USED TO STORE
C THE DERIVATIVE OF AQWZZ.
C
      REAL*8 TELZR,TELZD,ZMLZZ,TMLZZ,ZELZZ,
     1 TELZZ,ZHLZZ,ZKLZZ,ZYLOZ,ZYLDZ,ZYLMZ,
     2 ZYLSZ,ZYLWZ,ZPRZZ,ZULZZ,TEPSM,TEOSM,
     3 TMCCS,TMOMS,TMCSS,TMOWS,ZMDL,
     4 ZMML,ZMCL,ZMWL,ZMSL,AVLSM,WVLS,
     5 PPLR,TMOSM,TMPSM,TMUSM,TMDSM,TEUSM,
     6 EFLSM,BOXYS,TMCZF,TMWZF,TMSZF,TMMZF,
     + TMCZF,FQPCR,ZKCZZ,ZMOZZ,
     7 TMOZZ,TEOZZ,ZHPZZ,TMPZZ,
     8 TEPZR,ZRFZZ,ZHFZZ,FQLOR,
     9 FQWCR,TMOZB,PHI,AQWZZ,ZKWZZ,
     + FQLWR,FQPWR,FQLWD,
     1 TMNET,
     1 DZMLZZ,CZELZZ,CZMOZZ,DZMDL,DZMML,
     2 DZMOL,DZMWL,DZMSL,CAQWZZ,CZRFZZ
C
C
C NOTE THAT IF TELZR IS NOT THE FIRST VARIABLE IN COMMON /VAR/ THEN
C THE EQUIVALENCE STATEMENTS IN FUNCT, PACKX AND UNPACK MUST BE
C CHANGED.
      COMMON /VAR/ TELZR(MR),TELZD(MR),ZMLZZ(MR),TMLZZ(MR),ZELZZ(MR),
     1 TELZZ(MR),ZHLZZ(MR),ZKLZZ(MR),ZYLCZ(MR),ZYLDZ(MR),ZYLMZ(MR),
     2 ZYLSZ(MR),ZYLWZ(MR),ZPRZZ(MR),ZULZZ(MR),TEPSM(MR),TEOSM(MR),
     3 TMCDS(MR),TMOMS(MR),TMCSS(MR),TMOWS(MR),ZMCL(MR),
     4 ZMML(MR),ZMOL(MR),ZMWL(MR),ZMSL(MR),AVLSM(MR),WVLS(MR),
     5 PFLR(MR),TMCSM(MR),TMPSM(MR),TMUSM(MR),TMCSM(MR),TEUSM(MR),
     6 EFLSM(MR),BOXYS(MR),TMCZF(MR),TMWZF(MR),TMSZF(MR),TMMZF(MR),
     + TMOZF(MR),FQPCR(MC,MR),ZKCZZ(MO,MR),ZMCZZ(MO,MR),
     7 TMCZZ(MC,MR),TEOZZ(MC,MR),ZHPZZ(MC,MR),TMFZZ(MC,MR),
     8 TEPZR(MC,MR),ZRFZZ(MO,MR),ZHFZZ(MC,MR),FQLCR(MC,MR),
     9 FQWCR(MO,MR),TMOZB(MO,MR),PHI(MO,MR),AQWZZ(MW),ZKWZZ(2,MW),
     + FQLWR(2,MW),FQPWR(2,MW),FQLWD(2,MW),
*********************************************************************
     1 TMNET(MR),
*********************************************************************
     1 DZMLZZ(MR),DZELZZ(MR),DZMCZZ(MO,MR),DZMDL(MR),DZMML(MR),
     2 DZMOL(MR),DZMWL(MR),DZMSL(MR),DAQWZZ(MW),DZRFZZ(MC,MR)
```

```
C FILE BLDNG.CMN:
C
C THIS INCLUDE FILE CONTAINS ALL OF THE GLOBAL PARAMETERS WHICH
C DESCRIBE THE BUILDING IN WHICH THE FIRE IS CONTAINED.  IT CONTAINS
C SUCH INFORMATION AS ROOM, OBJECT, VENT AND WALL GEOMETRIES; PHYSICAL
C CONSTANTS FOR THE RUN, THE INTERCONNECTIONS OF ROOMS, WALLS AND VENTS;
C A DESCRIPTION OF THE NUMBER OF OBJECTS IN EACH ROOM, INCLUDING THE
C TYPE OF FIRE ON EACH OBJECT.
C
C      ZBVZZ -- WIDTH OF THE VENT.
C      ZHVZZ -- HEIGHT OF THE VENT (FROM BASE TO TRANSOM).
C      ZHTZZ -- DISTANCE FROM THE TRANSOM OF THE VENT TO
C               THE CEILING IN THE SAME ROOM.
C      CD -- VENT FLOW COEFFICIENT FOR ALL THE VENTS.
C      ZNWZZ -- THICKNESS OF THE WALL.
C      ZJWZZ -- THERMAL CONDUCTIVITY OF THE WALL.
C      ZGWZZ -- THERMAL DIFFUSIVITY OF THE WALL.
C      ZCWZZ -- SPECIFIC HEAT OF THE WALL.
C      VMWZZ -- DENSITY OF THE WALL MATERIAL.
C      ZOWZM -- MAXIMUM HEAT TRANSFER COEFFICIENT FOR WALLS.
C      ZOWZN -- MINIMUM HEAT TRANSFER COEFFICIENT FOR WALLS.
C      STATE -- THE BURNING STATE OF EACH OBJECT (E.G.
C               FLAMING, HEATING, ETC.)
C      ZXCZZ -- THE X-COORDINATE OF THE OBJECT.
C      ZYCZZ -- THE Y-COORDINATE OF THE OBJECT.
C      ZHCZZ -- THE HEIGHT OF THE OBJECT.
C      ZNCZZ -- THE THICKNESS OF THE OBJECT.
C      ZJCZZ -- THERMAL CONDUCTIVITY OF THE OBJECT.
C      ZGCZZ -- THERMAL DIFFUSIVITY OF THE OBJECT.
C      ZCCZZ -- SPECIFIC HEAT OF THE OBJECT.
C      VMCZZ -- DENSITY OF THE OBJECT.
C      EB -- EMMISIVITY/ABSORPTIVITY OF THE SURFACE OF THE OBJECT.
C      CHI -- FRACTION OF THE HEAT OF COMBUSTION RELEASED IN THE
C             BURNING OF THE OBJECT.
C      ZMCZO -- INITIAL FUEL MASS OF THE OBJECT.
C      ZRFZO --
C      ZRFZM -- MAXIMUM BURNING RADIUS.
C      QF --
C      QVAP -- HEAT OF VAPORIZATION OF THE OBJECT.
C      ZOAZZ --
C      ALPHA --
C      ZTIG -- TIME OF IGNITION OF THE OBJECT.
C      ZKCIG -- IGNITION TEMPERATURE OF THE OBJECT.
C      ZKCPY -- TEMPERATURE OF THE ONSET OF PYROLYSIS FOR THE OBJECT.
C      XGAMMA -- AIR/FUEL MASS RATIO IN A FREE BURN OF THE OBJECT.
C      XGAMAS -- STOICHIOMETRIC AIR/FUEL MASS RATIO FOR THE BURN-
C                ING OF THE OBJECT.
C      ZRFZI --
C      FCO2 -- MASS FRACTION OF CO2 EVOLVED IN THE PYROLYSIS
C              GASES OF THE OBJECT.
C      FCO -- MASS FRACTION OF CO EVOLVED IN THE PYROLYSIS
C             GASSES OF THE OBJECT.
C      FS -- MASS FRACTION OF SMOKE EVOLVED IN THE PYROLYSIS
C            GASES OF THE OBJECT.
C      FH2O -- MASS FRACTION OF WATER VAPOR EVOLVED IN THE PYROL-
C              YSIS GASES OF THE OBJECT.
C      ZKFZZ -- TEMPERATURE OF THE FLAMES OF THE OBJECTS.
C      ZLAMDA --
C      TMFGZ -- MASS FLOW RATE OF THE FUEL FOR THE OBJECT (IF THE
C               OBJECT IS A GASS BURNER.
C      ZUFZZ --
C      FIRTYP -- THE TYPE OF FIRE ON THE OBJECT (GROWING, POOL,
C               OR BURNER) IF THE OBJECT IS BURNING.
C     ANGV -- ANGLE OF THE SURFACE OF THE OBJECT WITH THE VERTICAL
C             PLANE THROUGH THE X-AXIS.
C     ANGH -- ANGLE OF THE SURFACE OF THE OBJECT WITH THE HORIZONTAL
C             PLANE.
C     AFIRE -- FIRE SPREAD PARAMETER FOR THE OBJECT.
C     PSIO -- INITIAL VALUE FOR THE SEMI-APEX ANGLE OF THE FLAME
C             OVER THE OBJECT.  THIS VALUE IS USED WHENEVER THE
C             FIRE OF THE OBJECT IS NOT OXYGEN STARVED.
C     TPSIO -- TANGENT OF THE SEMI-APEX ANGLE OF THE FLAMES OVER
C              THE OBJECT.  THIS VALUE IS USED WHENEVER THE FIRE
C              OF THE OBJECT IS NOT OXYGEN STARVED.
C     BETA --
C     OLDTIM -- THE TIME AT WHICH THE VALUES OF THE PREVIOUS
C               TIME STEP VALUES WERE TAKEN.
C     ZRFZZP -- RADIUS OF THE FIRE OF THE OBJECT AT THE PREV-
C               IOUS TIME STEP.
C     TMCZZP -- NEGATIVE OF MASS PYROLYSIS RATE AT THE PREVIOUS
C               TIME STEP.
C     ZHFZZP -- FLAME HEIGHT OF OBJECT AT THE PREVIOUS TIME STEP.
C     ZMCZZP -- MASS OF THE OBJECT AT THE PREVIOUS TIME STEP.
C     ZKCZZP -- TEMPERATURE OF THE UPPER SURFACE OF THE OBJECT
C               AT THE PREVIOUS TIME STEP.
C     ZLRZY -- LENGTH OF THE ROOM ALONG THE Y-DIMENSION.
C     ZLRZX -- LENGTH OF THE ROOM ALONG THE X-DIMENSION.
C     ZHRZZ -- HEIGHT OF THE ROOM.
C     ZKAZZ -- TEMPERATURE OF THE COLD LAYER IN THE ROOM.
C     VMAZZ -- DENSITY OF THE AIR IN THE COLD LAYER OF THE
C              ROOM.
C     FIROOM -- INDICATES IF THERE IS A FIRE IN THE ROOM.
C     VERSUN -- INTEGER ARRAY INDICATING WHICH VERSION OF
C               EACH PHYSICAL SUBROUTINE IS IN USE (ONLY FOR
C               THOSE PHYSICAL SUBROUTINES WITH MORE THAN ONE
C               VERSION.  CURRENTLY, VERSUN IS LARGE ENOUGH TO
C               HANDLE FIVE SUBROUTINES WITH MULTIPLE VERSIONS.
C     ZKAOUT -- TEMPERATURE OF THE AMBIENT AIR OUTSIDE THE BUILDING.
C     ZULOUT -- ABSORPTIVITY OF THE AIR OUTSIDE THE BUILDING.
C     VMAOUT -- DENSITY OF THE AIR OUTSIDE THE BUILDING.
C     CP -- SPECIFIC HEAT OF AIR.
C     YCO2 -- MASS CONCENTRATION OF CO2 IN THE OUTSIDE AIR.
C     YO2 -- MASS CONCENTRATION OF OXYGEN IN THE OUTSIDE AIR.
C     NROOMS -- NUMBER OF ROOMS IN THE BUILDING.
C     NWALLS -- NUMBER OF WALLS IN THE BUILDING.
C     NVENTS -- NUMBER OF VENTS IN THE BUILDING.
C     OBJS -- NUMBER OF OBJECTS IN THE ROOM.
C     WLINK(SIDE,WALL) -- THE ROOM ON THE "SIDE" OF THE "WALL".
C          NOTE THAT IF WLINK(SIDE,WALL)=0 THEN THE "SIDE" OF THE
C         "WALL" IS ON THE OUTSIDE OF THE BUILDING.
C     VLINK(SIDE,WALL) -- THE ROOM ON THE "SIDE" OF THE "VENT".
C          NOTE THAT IN VLINK(SIDE,VENT)=0 THEN THE "SIDE" OF THE
C          "VENT" IS ON THE OUTSIDE OF THE BUILDING.
C     LAYR(R) -- SET TO TRUE IF THERE IS A HOT LAYER IN THE ROOM.
C
C NAMES FOR THE VARIOUS FIRE STATES AND FIRE TYPES OF AN OBJECT
C (NOTE THAT COLD,HEATNG,CNSUM,FLAME ARE THE ONLY STATES USED IN
C THE CURRENT PROGRAM).
C   STATES:
C     COLD -- COLD.
C     HEATNG -- HEATNG.
C     PYROL -- PYROLYZING.
C     CHRCFR -- CHARCOAL FIRE.
C     EXTING -- EXTINGUISHED.
C     CNSUM -- CONSUMED.
```

```
C       SMOLD -- SMOLDERING.
C       FLAME -- FLAMING.
C
C   FIRE TYPES:
C       GROWNG -- GROWING.
C       POOL -- POOL.
C       BURNER -BURNER.
C
        INTEGER COLD,HEATNG,PYROL,CHRCFR,EXTING,CNSUM,SMOLD,FLAME,
     1          GROWNG,POOL,BURNER
C
        PARAMETER (COLD = 1, HEATNG = 2, PYROL = 3, CHRCFR = 8,
     1      EXTING = 9, CNSUM = 10, SMOLD = 4, FLAME = 5,
     2      BURNER = 3, GROWNG = 1, POOL = 2)
C
        REAL*8 ZBVZZ,ZHVZZ,ZHTZZ,CD
        REAL*8 ZNWZZ,ZJWZZ,ZGWZZ,ZCWZZ,VMWZZ,ZCWZM,ZOWZN
        REAL*8 ZXOZZ,ZYOZZ,ZHOZZ,ZNOZZ,ZJOZZ,ZGOZZ,ZCOZZ,VMOZZ,EB,
     1      CHI,ZMOZO,ZRFZO,ZRFZM,QF,QVAP,ZCAZZ,ALPHA,ZTIG,ZKOIG,
     2      ZKOPY,XGAMMA,XGAMAS,ZRFZI,FCO2,FCO,FS,FH2O,ZKFZZ,
     3      ZLAMDA,TMFGZ,ZLFZZ,ANGV,ANGH,AFIRE,PSIO,TPSIO,BETA
        INTEGER STATE,FIRTYP
        REAL*8 CLDTIM,ZRFZZP,TMOZZP,ZHFZZP,ZMOZZP,ZKOZZP
        REAL*8 ZLRZY,ZLRZX,ZHRZZ,ZKAZZ,VMAZZ
        LOGICAL FIRCOM,LAYR
        INTEGER VERSUN
        REAL*8 ZKAOUT,ZULOUT,VMAOUT,CP,YCO2,YO2
        INTEGER NROOMS,NWALLS,NVENTS,OBJS,WLINK,VLINK
        PARAMETER (ZULOUT = .001D0,YCO2=.0005D0,YO2=.2318D0)
        COMMON /BLDNG/ ZBVZZ(MV),ZHVZZ(MV),ZHTZZ(MV),CD,ZNWZZ(MW),
     + ZJWZZ(MW),ZGWZZ(MW),ZCWZZ(MW),VMWZZ(MW),ZCWZM,ZOWZN,
     + STATE(MO,MR),ZXOZZ(MO,MR),ZYOZZ(MO,MR),ZHOZZ(MO,MR),
     + ZNOZZ(MO,MR),ZJOZZ(MO,MR),ZGOZZ(MO,MR),ZCOZZ(MO,MR),
     + VMOZZ(MO,MR),EB(MO,MR),CHI(MO,MR),ZMOZO(MO,MR),
     + ZRFZO(MO,MR),ZRFZM(MO,MR),QF(MO,MR),QVAP(MO,MR),ZCAZZ,
     + ALPHA,ZTIG(MO,MR),ZKOIG(MO,MR),ZKOPY(MO,MR),XGAMMA(MO,MR),
     + XGAMAS(MO,MR),ZRFZI(MO,MR),FCO2(MO,MR),FCO(MO,MR),
     + FS(MO,MR),FH2O(MO,MR),ZKFZZ,ZLAMDA,TMFGZ(MO,MR),ZLFZZ,
     + FIRTYP(MO,MR),ANGV(MO,MR),ANGH(MO,MR),AFIRE(MO,MR),
     + PSIO(MO,MR),TPSIO(MO,MR),BETA(MO,MR),CLDTIM,ZRFZZP(MO,MR),
     + TMOZZP(MO,MR),ZHFZZP(MO,MR),ZMOZZP(MO,MR),ZKOZZP(MO,MR),
     + ZLRZY(MR),ZLRZX(MR),ZHRZZ(MR),ZKAZZ(MR),VMAZZ(MR),
     + FIRCOM(MR),VERSUN(5),ZKAOUT,VMAOUT,CP,NROOMS,NWALLS,
     + NVENTS,OBJS(MR),WLINK(2,MW),VLINK(2,MV),LAYR(MR)
```

```
C FILE DIMN.INC
C
C THIS INCLUDE FILE CONTAINS INFORMATION ON THE DIMENSIONS OF
C ARRAYS USED IN THE INTERFACE.
C
C      MAXRMS -- MAXIMUM NUMBER OF ROOMS ALLOWED.
C      MR -- A SYNONYMN FOR MAXRMS.
C      MAXOBS -- MAXIMUM NUMBER OF OBJECTS ALLOWED IN EACH ROOM.
C      MO -- A SYNONYMN FOR MAXOBS.
C      MAXVTS -- MAXIMUM NUMBER OF VENTS ALLOWED.
C      MV -- A SYNONYM FOR MAXVTS.
C      MAXWLS -- MAXIMUM NUMBER OF WALLS ALLOWED.
C      MW -- A SYNONYMN FOR MAXWLS.
C NOTE THAT THE ABOVE SYNONYMNS ARE NEEDED SO THAT THE COMMON
C BLOCK DECLARATIONS IN /BLDNG/ AND /VAR/ WILL NOT EXCEED THE
C MAXIMUM NUMBER OF ALLOWABLE CONTINUATION LINES.
C      RMVARS -- NUMBER OF ARRAYS IN SECTION 1 OF /VAR/ INDEXED
C                 XXXXX(MAXRMS).
C      OBVARS -- NUMBER OF ARRAYS IN SECTION 1 OF /VAR/ INDEXED
C                XXXXX(MAXOBS,MAXRMS).
C      VTVARS -- NUMBER OF ARRAYS IN SECTION 1 OF /VAR/ INDEXED
C                XXXXX(MAXVTS).
C      WLVARS -- NUMBER OF ARRAYS IN SECTION 1 OF /VAR/ INDEXED
C                XXXXX(MAXWLS).
C      SDVARS -- NUMBER OF ARRAYS IN SECTION 1 OF /VAR/ INDEXED
C                XXXXX(2,MAXWLS).
C      NVARS -- SIZE OF SECTION 1 OF COMMON /VAR/.
C      VARSIZ -- THE NUMBER OF VARIABLES IN ALL OF COMMON /VAR/
C                (INCLUDING SECTIONS 2 AND 3).
C      MAXVRS -- THE MAXIMUM NUMBER OF VARIABLES WHICH ARE ALLOWED
C                IN THE SYSTEM.
C
      INTEGER MAXRMS,MAXOBS,MAXVTS,MAXWLS,MR,MO,MV,MW,MAXVRS,VARSIZ
C
      PARAMETER (MAXRMS = 5, MAXOBS = 3, MAXVTS = 6, MAXWLS = 5)
      PARAMETER (MR = MAXRMS, MO = MAXOBS, MV = MAXVTS, MW = MAXWLS)
C
      INTEGER RMVARS,OBVARS,VTVARS,WLVARS,NVARS,SDVARS
C
      PARAMETER (RMVARS = 41, OBVARS = 14, VTVARS = 0,
     1     WLVARS = 1, SDVARS = 4, MAXVRS = 200,
     2     NVARS = MAXRMS*RMVARS + MAXOBS*MAXRMS*OBVARS +
     3          MAXVTS*VTVARS + 2*MAXWLS*SDVARS + MAXWLS*WLVARS,
     4     VARSIZ = NVARS + 8*MR + 2*MO*MR + 1*MW)
```

```
C FILE DSCNAM.INC
C
C THIS FILE CONTAINS THE SYMBOLIC NAMES FOR THE VARIOUS TYPES OF
C UNPLANNED DISCONTINUITY WHICH MAY OCCUR IN THE COURSE OF A RUN.
C
C BRNOUT -- THE BURNOUT OF AN OBJECT.
C HEATIG -- IGNITION OF AN OBJECT BY HEATING.
C CNTCIG -- IGNITION OF AN OBJECT BY CONTACT WITH THE FLAMES OF ANOTHER
C           OBJECT.
C STRTUP -- THE START OF A RUN.
C
      INTEGER BRNOUT,HEATIG,CNTCIG,STRTUP
C
      PARAMETER (BRNOUT = 1, HEATIG = 2, CNTCIG = 3, STRTUP = 4)
```

```
C FILE IC.INC
C
C THIS FILE CONTAINS THE CHANNEL NUMBERS FOR INPUT AND OUTPUT TO
C THE DISK AND TO THE TERMINAL SCREEN.
C
C IWTTY -- CHANNEL NUMBER FOR WRITES TO THE TERMINAL SCREEN.
C IRTTY -- CHANNEL NUMBER FOR READS FROM THE TERMINAL SCREEN.
C IWDSK -- CHANNEL NUMBER FOR WRITES TO THE DISK FILE FIRE.DAT.
C
      INTEGER IWTTY,IRTTY,IWDSK
C
      PARAMETER (IWTTY = 6, IRTTY = 5, IWDSK = 19)
```

NBS-114A (REV. 9-78)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBS-GCR-83-451	2. Gov't. Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

Computer Fire Code VI - Vol. 2

5. Publication Date

December 1983

6. Performing Organization Code

7. AUTHOR(S)

J.B. Gahm

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Harvard University
Division of Applied Sciences
Cambridge, MA

10. Project/Task/Work Unit No.

11. Contract/Grant No.

NB82NADA3030

12. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

National Bureau of Standards
Department of Commerce
Washington, DC 20234

13. Type of Report & Period Covered

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

This is the documentation for Computer Fire Code VI (CFC VI). There is no definitive version of CFR VI yet, but this document will be valid for any version. The numerical "package" used is described in general terms only. The physics inherent in the program is also omitted; however, most of those subroutines have been taken directly from Mark 5, although they have been broken up so that any subroutine has just one output. For the reader interested in the documentation of the physics, it is given in detail in Harvard Fire Project Technical Reports 34 and 45.

There are four appendices, wherein it is explained how to insert a new subroutine or a new physical variable. One of the appendices is a dictionary of symbols, terms, and variables used in the program and in the text. Appendix D is in a separate volume.

Following the report, a listing of the program appears; this is a version of CFC VI which compiles on both a VAX and a Perkin-Elmer computer. The program was written in ANSI 77 FORTRAN, and is fairly machine-independent. It consists of two independent parts: BINP is a program which produces an input file for the second (main) program, DBLE, to use. Two differences between CFC VI and Mark 5 not described in the Foreword are: first, the gas burner algorithm in Mark 5 has not been incorporated. Second, the gas concentrations are incorrectly computed.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

compartment fires; computer programs; fire models; fortran

18. AVAILABILITY ☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☐ Order From Sup. of Doc., U.S. Government Printing Office, Washington, DC 20402, SD Stock No. SN003-003-

☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

19. SECURITY CLASS (THIS REPORT)

UNCLASSIFIED

20. SECURITY CLASS (THIS PAGE)

UNCLASSIFIED

21. NO. OF PRINTED PAGES

22. Price

USCOMM-DC